

MODELING OF SOLID-SUPERCRITICAL FLUID EQUILIBRIA USING CUBIC EQUATIONS OF STATE: A UNIFIED APPROACH

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Technology

by

B. Chender Rao

to the

Department of Chemical Engineering
Indian Institute of Technology, Kanpur

February 1996.

22 MAR 1996

CENTRAL LIBRARY
I. I. T., KANPUR

Acc. No. A. 121229

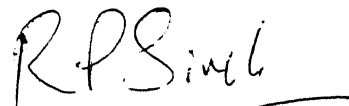
CHE- 1996- M- RAO- MOD



A121229

CERTIFICATE

It is certified that the work contained in the thesis titled **Modeling of Solid-Supercritical Fluid Equilibria Using Cubic Equations of State: A Unified Approach**, by *B. Chender Rao*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.



Dr. R.P. Singh

Professor

Department of Chemical Engg.

IIT Kanpur

22 February, 1996

Acknowledgement

I express my deepest sense of gratitude and appreciation for my thesis supervisor Dr. R.P. Singh for his guidance, encouragement, criticisms and advice made during this work.

It is a pleasure to thank my friends and classmates who made the stay at IIT Kanpur a memorable one.

My special thanks to the faculties and authorities of IIT Kanpur who make this institution an excellent learning place.

I take this opportunity to thank Engineers India Ltd., for the financial support provided by them during this work.

B. Chender Rao

Contents

Abstract	iv
List of Figures	vi
List of Tables	vii
Nomenclature	viii
1 Introduction	1
2 Literature Survey	4
3 Thermodynamic Modeling For Solid-Fluid Equilibria	10
3.1 Equation of State Modeling of Solubilities	10
3.2 Cubic Equations-of-State	11
3.3 Fugacity Coefficient from Cubic EOS	12
3.4 A Simplified Cubic EOS Model for Solubility	14
4 Results and Discussion	17
4.1 Physical Properties	17
4.2 Data Reduction and Parameter Estimation Criteria	19

4.3 Results and Discussion	20
4.4 Conclusions	61
 Bibliography	 62
 Appendix	 69
 A Program Listing of the Rigorous Model	 70
 B Program Listing of the Simplified Model	 93

Abstract

The appropriate mathematical modeling of the thermodynamic behaviour of the solute-solvent system is important for effective design of the extraction and separation units of a supercritical fluid extraction (SFE) process. In the present work, several combinations of cubic equations of state (EOS) and mixing rules have been studied, computing binary interaction parameters (BIPs) and absolute average errors in solubility for binary systems of 14 solid solutes and 5 supercritical solvents at different temperatures. Equations of state chosen for this study were Redlich-Kwong (RK), Soave (SRK) and Peng-Robinson (PR), whereas the mixing rules chosen were quadratic and cube-root using a single BIP, k_{12} or two BIPs, k_{12} and l_{12} . As expected, the introduction of the second BIP, l_{12} , greatly improves the accuracy of the calculated solubility values. The results are highly satisfactory, considering the uncertainty in experimental measurements of low solubility values as well as the uncertainty of the T_c , P_c and ω values of the solute. The comparison of overall percent average errors indicate that, globally, no distinction can be made either among the EOS or the mixing rules. However, differences among the EOS are found when the errors for individual systems are considered. Promising results have been obtained also using a recently proposed simplified model with each of the three cubic EOS studied. In this method the infinite-dilution fugacity coefficient of the solute in the supercritical fluid phase is computed using the infinite-dilution compressibility factor, and the EOS parameters α and β , defined by $\alpha = \sqrt{a_2/a_1}$ and $\beta = b_2/b_1$, are treated as adjustable parameters; the conventional iterative procedure is thus substituted by a once-through calculation. Optimum solute-to-solvent parameter ratios α and β and percent average errors have been computed for binary systems of 15 solid solutes and 5 solvents at different temperatures. The simplified method provides an efficient and easy way to correlate the solubilities of high-molecular-weight solids in supercritical

fluids; since no critical properties of the solute are required.

The unified approach adopted in the rigorous EOS modeling of solid-supercritical fluid equilibria integrates different two-parameter cubic EOS with different mixing rules in a cohesive manner. It should be especially advantageous in testing the capabilities of various multiparameter cubic EOS/mixing rule combinations in correlating and predicting solid-supercritical fluid equilibria.

List of Figures

4.1	Calculated Solubilities of Anthracene in Supercritical CO ₂ at 323.15 K Using the PR EOS/QMR formulation: Role of Binary Interaction Parameters	23
4.2	Calculated Solubilities of Benzoic Acid in Fluoroform at 328.25 K Using Different EOS/CMR Formulation with Optimum k_{12}	24
4.3	Calculated Solubilities of Naphthalene in Fluoroform at 328.25 K Using Different EOS/CMR Formulation with Optimum k_{12} & l_{12}	24
4.4	Calculated Solubilities of Fluorene in Ethylene at 343.15 K Using Different EOS/QMR Formulation with Optimum k_{12}	25
4.5	Calculated Solubilities of Benzoic Acid in CO ₂ at 338 K Using Different EOS/QMR Formulation with Optimum k_{12} & l_{12}	25
4.6	Calculated Solubilities of Benzoic Acid in Supercritical CO ₂ at Different Temperatures Using the Simplified Model with the PR EOS	26
4.7	Calculated Solubilities of Phenanthrene in Supercritical CO ₂ at Different Temperatures Using the Simplified Model with the PR EOS	27
4.8	Calculated Solubilities of Acridine in Supercritical CO ₂ at Different Temperatures Using the Simplified Model with the PR EOS	28
4.9	Calculated Solubilities of 1,4-Naphthaquinone in Supercritical Ethane at Different Temperatures Using the Simplified Model with the PR EOS . .	29

List of Tables

3.1	Cubic Equations of State	11
4.1	Physical Properties of the Supercritical Solvents	17
4.2	Physical Properties of the Solid Solutes	18
4.3	Sublimation Pressures of Polychlorinated Biphenyls	19
4.4	EOS Modeling of Solubility: Optimum k_{12} Using the Quadratic Mixing Rule (QMR)	30
4.5	EOS Modeling of Solubility: Optimum k_{12} Using the Cube Root Mixing Rule (CMR)	36
4.6	EOS Modeling of Solubility: Optimum k_{12} & l_{12} Using the Quadratic Mixing Rule (QMR)	42
4.7	EOS Modeling of Solubility: Optimum k_{12} & l_{12} Using the Cube Root Mixing Rule (CMR)	48
4.8	EOS Modeling of Solubility: Optimum Solute to Solvent Parameter Ratios α & β in the Simplified Method	54

Nomenclature

a	mixture cohesive energy parameter in cubic EOS
a_{12}	unlike cohesive energy parameter in the mixing rules
b	mixture repulsive energy parameter in cubic EOS
b_{12}	unlike repulsive energy parameter in the mixing rules
f_i^g	fugacity of component i in fluid phase
f_i^s	fugacity of component i in solid phase
k_{12}	binary interaction parameter in the mixing rule for a
l_{12}	binary interaction parameter in the mixing rule for b
M_i	molecular mass of component i
n_i	mole numbers of component i
P	pressure
P_2^s	vapor pressure of the solid
P_c	critical pressure
R	universal gas constant
T	temperature
T_c	critical temperature
u	SW EOS constant
v	molar volume
v^s	solid volume
w	SW EOS parameter
y_i	mole fraction of component i in the fluid phase

y_2	solubility (mole fraction) of the solid solute in the supercritical phase
$y_{2,exp}$	experimental solubility
$y_{2,cal}$	calculated solubility
z	compressibility factor of the mixture

Greek Letters

α	an EOS parameter (Table 3.1); a parameter in simplified cubic EOS model = $\sqrt{a_2/a_1}$
β	a parameter in simplified cubic EOS model = b_2/b_1
Δ	defined as $\sqrt{u^2 - 4w}$
ϕ_2^{sat}	fugacity coefficient of the pure solid at saturation conditions
ϕ_2	fugacity coefficient of the solid in the supercritical phase
Ω_a	a cubic EOS dependent parameter listed in Table 3.1
Ω_b	a cubic EOS dependent parameter listed in Table 3.1

Subscripts

cal	calculated
exp	experimental
i	i^{th} component

Superscripts

g	gas
s	solid

Abbreviations

AE(%)	Percentage Average Error
BIP	Binary Interaction Parameter
CMR	Cube-root Mixing Rule
DCB	Dichlorobiphenyl
DMN	Dimethylnaphthalene

Abbreviations (contd.)

EOS	Equations of State
HCB	Hexachlorobiphenyl
HMB	Hexamethylbenzene
HMW	High Molecular Weight
M.W.	Molecular Weight
N	No. of Experimental Data Points
NA	Not Available
OF	Objective Function
PCBs	Polychlorinated Biphenyls
PR	Peng-Robinson
QMR	Quadratic Mixing Rule
RK	Redlich-Kwong
SFE	Supercritical Fluid Extraction
SRK	Soave-Redlich-Kwong
SW	Schmidt-Wenzel
TCB	Trichlorobiphenyl
TPM	Triphenylmethane
vdW	van der Waals

Chapter 1

Introduction

Supercritical fluid extraction technology has been under accelerated development in recent years. A myraid of potential application of supercritical fluid extraction (SFE) has emerged in disciplines such as biotechnology, food processing , environmental control and pharmaceutical and petroleum industry (Brennecke and Eckert, 1989; Johnston and Peck, 1989; Larson and King, 1986). Several fundamental aspects of the problem have to be looked into and solved before the concept reaches the industrial arena. These fundamental aspects consist of thermodynamics; including solubility measurements and modeling; transport phenomena; including measurements and modeling of transport properties (viscosity, thermal conductivity and diffusion coefficient) and transport coefficients (heat and mass transfer coefficients).

In many of the applications, especially in biotechnology and food industry, a supercritical fluid is used to extract or dissolve high-molecular-weight (HMW) compounds. These are usually solid at room conditions and the extent to which they dissolve in the supercritical fluid is small. The design of any particular SFE process is based upon the knowledge of the solubilities of the low-volatile components in the supercritical solvents. One of the major factors limiting the commercial success of SFE is the lack of extensive, reliable solubility data for the efficient design of separation units or the development of new methods for correlating and predicting supercritical fluid phase equilibria.

Correlation models for the solubilities of solids in supercritical fluids have been presented by many investigators, including the application of cubic equations of state

(EOS). The key point of the EOS method is the choice of a proper mixing rule to determine the mixture parameters. Usually the conventional van der Waals (vdW) mixing rules are used with empirically adjusted binary interaction parameters. The binary parameters cannot be correlated, and even with the optimal parameters, appreciable errors still exist for various cubic EOS (Haselow et al., 1986). This failure may be a result of the uncertainty in either the pure compound properties or in the solubility data. The experimental solubility data have an error which is often quite significant. For example, Johnston et al. (1982) report an experimental error higher than 40% for the lower end (below 10^{-5} mole fraction) of solubility data. In such cases, increased correlation errors are expected.

Recent work in the use of cubic EOS to represent solid-supercritical fluid equilibria has focused on the need to develop new mixing rules for highly asymmetric mixtures which comprise the majority of the solid-supercritical fluid systems of interest. However, the comparison between some recent correlative models using cubic EOS with new mixing rules, such as those of Sheng et al. (1992) and Spiliotis et al. (1994), with the standard one-fluid vdW mixing rules suggests that there is no significant advantage of one method over the other. Any difference is usually within the uncertainty involved in measuring small solubility values. When dealing with HMW solids, therefore, simpler models are usually of more practical interest since the solubility is small.

The lack of reliable critical data for many solid solutes of interest, especially complex organic molecules, poses another problem when using cubic EOS to model solid-supercritical fluid systems. Since these are needed for conventional EOS modeling, they are commonly estimated using conventional method. Then, the EOS-based model has to be fine-tuned by including binary interaction parameters (BIPs) to improve the quality of the fit. The BIPs so calculated will be highly dependent on the method used to estimate the solute's critical properties. The use of estimation methods puts an additional burden on the BIPs since they must correct not only the inadequacies of the EOS, but also the uncertainties introduced by the estimation methods. Recently Estévez et al. (1994) proposed a simplified model using the Peng-Robinson (PR) EOS with standard vdW mixing rules to correlate the solubility of solids in supercritical fluids. In this method the EOS parameters for the solute are taken as adjustable parameters rather than attempting to estimate them

based on critical properties. The proposed method is considerably simpler than the conventional (rigorous) EOS approach and is especially advantageous for correlating solubility of HMW solids in supercritical fluids, since it does not require the values of critical temperature or critical pressure of the solute.

In the present work, several combinations of cubic EOS and mixing rules have been studied, computing binary interaction parameters (BIPs) and absolute average errors in solubility for binary systems of 14 solid solutes and 5 supercritical solvents at different temperatures. Equations of state chosen for this study were Redlich-Kwong (RK), Soave (SRK) and Peng-Robinson (PR), whereas the mixing rules chosen were quadratic (van der Waals) and cube-root using a single BIP, k_{12} , or two BIPs, k_{12} and l_{12} . As expected, the introduction of the second BIP, l_{12} , greatly improves the accuracy of the calculated solubility values. From the overall percent average errors, we can conclude that the cube-root mixing rule performs as well as the conventional vdW mixing rule in representing solid-supercritical fluid equilibria and, in general, no distinction can be made among the EOS. However, differences among the EOS are found when the errors for individual systems are compared. The results of correlation using two BIPs are highly satisfactory, considering the uncertainty in experimental measurements of low solubility values as well as the uncertainty of the T_c , P_c and ω values of the solute. Promising results have been obtained also using the simplified model of Estévez et al. (1994) with each of three EOS (RK, SRK and PR) studied. Optimum solute-to-solvent parameter ratios α and β and percent average errors have been computed for binary systems of 15 solid solutes and 5 solvents at different temperatures. The simplified method provides an efficient and easy way to correlate HMW solids solubilities in absence of the critical properties of the solute.

The unified approach adopted in the rigorous EOS modeling of solid-supercritical fluid equilibria integrates different two-parameter cubic EOS with different mixing rules in a cohesive manner. It should be especially advantageous in testing the capabilities of various multiparameter cubic EOS/mixing rule combinations in correlating and predicting solid-supercritical fluid equilibria.

Chapter 2

Literature Survey

Supercritical fluid extraction (SFE) is a relatively new separation technique that has received much attention in recent years (McHugh and Krukonis, 1994). Conventional distillation methods involve higher temperatures which result in decomposition of heat-labile substances. Conventional extraction methods involve chemical solvents which, even in residual quantities, are of concern from the standpoint of toxicity to consumers of the end products. Supercritical extraction, however, is performed at relatively low temperatures using solvents which can be easily separated from the end products by simple physical means and is a new alternative to distillation or liquid extraction (Parkinson and Johnson, 1989). Among the supercritical solvents, carbon dioxide is usually preferred, since it is nontoxic, nonflammable, environmentally acceptable and relatively inexpensive, and its low critical temperature (31.2 °C) allows extraction at relatively low temperatures convenient for heat-labile substances. SFE exploits the advantage of a solubility enhancement of 10^3 - 10^8 in supercritical fluids and the phenomenon of enhanced solubility in a near-critical fluid of substances which are solid at the temperature of the system is well-documented (Robin and Vodar, 1953; Paulaitis et al., 1983; Johnston and Penninger, 1989; Bruno and Ely, 1991). This enhanced solubility is related primarily to the "liquid-like" density of the fluid which promotes strong attractive forces. Additional attractive features of SFE are that the low viscosity of the supercritical fluid combined with high solute diffusivities result in superior mass transfer characteristics. Furthermore, significant density gradients between the particle-fluid interface and the bulk supercritical fluid, which are caused by buoyant forces, result in enhanced mass transfer. This enhanced

mass transfer is in contrast to the traditional liquid solvent extraction, which generally takes several hours to perform, requires relatively large amounts of solvent, and may result in incomplete recovery. In comparison to the conventional processes, SFE offers considerable flexibility for an extractive separation through controlling pressure, temperature, and choice of solvent (Todd and Elgin, 1955). Its main advantage over the conventional ones is that the dissolved extract may be completely separated from the supercritical fluid simply by decreasing the fluid density, thus its dissolving power. This may be accomplished by expansion to subcritical pressure or by a relatively small increase in temperature. The energy requirements may thus be reduced significantly for volatilization and separation of the heavy component (Kohn and Savage, 1979). SFE has already been applied in a number of processes such as decaffeination of coffee, production of enzymes and other pharmaceutical substances. SFE is indeed very tempting as a separation process in the presence of thermosensitive substances. Recent studies have demonstrated the use of supercritical fluids also in environmental control, e.g. for extraction and removal of toxic organic contaminants from hazardous wastes and various solids such as sediments and fly ash (Brady et al., 1987; Paulaitis et al., 1987; Dooley et al., 1987, 1990). Extraction and recovery of aromatic hydrocarbons from contaminated soils and waters by means of supercritical fluids have been investigated by Hawthorne and Miller (1987), Roop et al. (1989), and Akgerman et al. (1992).

Although the number of applications using supercritical fluids is increasing constantly, their commercial success is still limited. Solubility dependence of the extracted components on pressure and temperature forms the basis of design for conditions in the extractor and separator. One of the major factors limiting the commercial success of SFE is the lack of extensive, reliable solubility data for the efficient design of separation units or the development of new methods for correlating and predicting high-pressure and supercritical fluid phase equilibria. The development of new applications for supercritical fluids increasingly implies in-depth knowledge of solute-solvent systems, the complexity of which is due to the highly non-ideal behaviour of the solutions. The appropriate mathematical modeling of the thermodynamic behaviour of the solute-solvent system is important for effective design of the extraction and separation units of a SFE process. Some current treatments attempt empirical relations (Chrastil, 1982; Adachi and Lu, 1983; Schmitt and Reid, 1985; Kramer and Thodos, 1988; Iwai et al., 1991; Sakaki, 1992; Weng and Lee, 1992; and Politzer et

al., 1993) or do modeling that focuses on the virial equation of state (Ewald et al., 1953; Hinckley and Reid, 1964; Chueh and Prausnitz, 1967; Quiram et al., 1994) and cubic equations of state (Dobbs and Johnston, 1987; Dobbs et al., 1987; Kosal et al., 1992; Caballero and Estévez, 1991, 1992; Caballero et al., 1992; Mukhopadhyay and Rao, 1993; Yu et al., 1995; Coutikos et al., 1995; Chen et al., 1995). Even more complex equations of state (EOS) have been used to represent fluid-phase nonideality in correlating solid-fluid equilibria (Finck et al., 1992; Bamberger et al., 1994; Bamberger and Maurer, 1994; Deak and Kemeny, 1994; Barna et al., 1994). Since experimental measurements, at pressures as high as those involved in SFE, are rather difficult, a model for prediction of solubilities is very desirable for design purposes. However, recent attempts at the prediction of solubility of aromatic solids in supercritical CO₂ with a group-contribution equation of state (Bamberger and Maurer, 1994; Bamberger et al., 1994) or with the use of predictive excess Gibbs free energy (G^E) models in the attractive term parameter of cubic EOS (Spiliotis et al., 1994) – the so called predictive EOS/ G^E models – has not met with much success; the errors being typically in the range of 20-50%.

The empirical correlations, which are often described as linear relations, rarely show the behaviour of enhanced solubility of solids in supercritical fluids over wide ranges of pressure (Knez and Steiner, 1992). Recently, a linear relationship has been developed (Wang and Tavlarides, 1994) based on a dilute two-region solution theory. This relationship has the capability to quantitatively describe the solubility behaviour of a heavy solute in a compressed gaseous solution. Yu et al. (1995) obtained a good correlation of the linear equation to the experimental solubility data of polychlorinated biphenyls (PCBs) with the average deviations ranging from 11 to 19%. A problem often faced with the correlation – and of course prediction – of solid solubilities in supercritical gases is the lack of vapor pressures of solutes (e.g. solid n-alkanes). In such cases, a solution model, based on the assumption that the supercritical phase is dense enough to behave like a (hypothetical) liquid (expanded liquid treatment), that describes solid-expanded liquid equilibria has been applied to correlate the solubility data (Iwai et al., 1992; Spiliotis et al., 1994). It has the advantage that only the melting point of the solute and its enthalpy of melting, which are usually available, are needed. In general, however, cubic EOS or even EOS derived from statistical mechanics and lattice gas models have been applied in the correlation of solubility data with good results. Since EOS derived from lattice gas models are

more complicated than a simple cubic EOS (e.g. Johnston et al., 1982; Balbuena et al., 1991), we focus our attention upon the cubic ones.

Cubic EOS e.g. Redlich and Kwong (1949), Soave (1972) and Peng and Robinson (1976) have been widely employed to correlate solubilities of solids in supercritical fluids. These EOS and their temperature-dependent functions were originally derived to describe VLE systems. It has been suggested that these EOS may not be appropriate to describe systems in which solid solutes are involved. As pointed out by Schmitt and Reid (1986), the use of these equations to deal with solid solutes implicitly assumes the solid phase to be a subcooled liquid, and as such, the integration step used when equating fugacities proceeds essentially from a vapor to a liquid phase. In spite of this, these equations have been found to correlate binary solid-supercritical fluid experimental data quite well in region away from criticality, especially data for non-polar symmetric systems. Lee et al. (1988) examined the Redlich-Kwong (RK), the Soave (SRK), and the Peng-Robinson (PR) EOS to evaluate their capability for correlating solubility data of solids in compressed gases. They correlated available experimental data for 35 binary systems using both one and two binary interaction parameters (BIPs) in the conventional van der Waals (vdW) mixing and combining rules to express the composition dependence of EOS parameters. Results were reported in terms of one average BIP or one pair of average BIPs applicable to a range of experimental temperatures and pressures. This way of correlating BIP values might not be appropriate if these values are intended for predictive purposes. It is well-known that at least the BIP used in the cohesive energy parameter is temperature dependent and this dependency is strongest for those solid compounds with low melting points and/or solubilities above the mole fraction of 0.01 at the highest temperatures and pressures (Bartle et al., 1991). Recent work in the use of cubic EOS to represent solid-supercritical fluid equilibria has focused on the need to develop new mixing rules for highly asymmetric mixtures which comprise the majority of the solid-supercritical fluid systems of interest and to cope with the limitations imposed by the use of conventional corresponding states theory based on critical properties (Rao and Mukhopadhyay, 1988; Soave, 1991; Spiliotis et al., 1994). Rao and Mukhopadhyay (1988) proposed a covolume dependent mixing rule for the cohesive (attraction) energy parameter of the PR and SRK EOS. Results show a reduction in the sensitivity of the calculated solubility values to the BIP. Spiliotis et al. (1994) employed the cubic t-mPR EOS (Magoulas and Tassios, 1990) to correlate the solubility of aromatic

solids in CO₂ using the conventional one-fluid vdW mixing rules with either one or two BIPs and also the EOS/G^E model based on the UNIQUAC G^E expression. In addition, an EOS/G^E model which coupled the SRK EOS with UNIQUAC was also investigated. Very good results were obtained with the t-mPR EOS using either the EOS/G^E model or two BIP conventional mixing rules, with the latter possessing a slight advantage. The comparison between correlative models, such as those of Sheng et al. (1992) (modified EOS/G^E), Johnston et al. (1982) (augmented vdW), and the t-mPR EOS models suggests that there is no significant advantage of one method over the other. Any difference is usually within the uncertainty of the experimental data and cannot be considered significant. Chen et al. (1995) proposed a modified mixing model with a volume correction term to calculate the solid solubilities in supercritical CO₂ and cosolvents using the PR EOS coupled with the modified UNIFAC group contribution G^E model (Larsen et al., 1987; Sheng et al., 1989). This volume correction term is well-correlated as a function of the molar volume of the solid molecule. This mixing model can be applied to various EOS with the same type of correlations for the volume correction constants (Chen et al., 1994). With this mixing model, solid solubilities in supercritical fluids can be predicted with acceptable accuracy.

The lack of reliable critical data for many solid solutes of interest, especially complex organic molecules, poses another problem when using cubic EOS to model solid-supercritical fluid systems. The usual approach has been to use several estimation methods (Reid et al., 1987). The use of estimation methods puts an additional burden on the BIPs since they must correct not only the inadequacies of the EOS, but also the uncertainties introduced by the estimation methods. To avoid the use of estimated properties, Schmitt and Reid (1986) used a modified Peng-Robinson EOS in which the pure component parameters for the solid solutes were treated as variables, and no BIPs were used in the conventional vdW mixing rules. The modified PR EOS with its two adjustable parameters for the solute correlated the solubility data better than the classical PR EOS with a single BIP. Recently Estévez et al. (1994) proposed a simplified method using the PR EOS with standard vdW mixing rules to correlate the solubility of solids in supercritical fluids. It is especially advantageous for correlating solubility of high-molecular-weight solids, since it does not require the critical properties of solutes.

Recent interest in supercritical fluid research has centered on the behaviour of

supercritical fluid mixtures near critical points. The formation of macroscopic clusters of solvent about the solute has been observed to occur at points of criticality resulting in an increase in solubility of the solute in the solvent (Kim and Johnston, 1987). Conventional cubic EOS are incapable of predicting this kind of behaviour. More complex models have been developed taking into account the clustering phenomenon. An excellent review on general modeling of supercritical mixtures, including the use of cubic EOS, has been made by Johnston et al. (1989). In spite of their limitations, cubic EOS exhibit a good simplicity/accuracy balance which makes them particularly attractive for engineering purposes.

Chapter 3

Thermodynamic Modeling For Solid-Fluid Equilibria

3.1 Equation of State Modeling of Solubilities

At solid-fluid phase equilibrium, the condition of equal fugacities in both phases (s,g)

$$f_i^s = f_i^g \quad (3.1)$$

is fulfilled for each component i . With the usual assumptions: the solubility of the supercritical fluid solvent (1) in the solid phase is negligible and the solid phase is pure; the solid phase is incompressible; the fugacity coefficient of the pure solid component (2) at saturation conditions, ϕ_2^{sat} , is taken to be unity since the vapor pressure of solid is very low, the fugacity f_2^s is given by

$$f_2^s = P_2^s \exp \left[\frac{v_2^s (P - P_2^s)}{RT} \right] \quad (3.2)$$

where P_2^s is the vapor pressure of the solid at temperature T , v_2^s is the solid molar volume, and P represents the system pressure. The fugacity of the solute (2) in the fluid phase is given by

$$f_2^g = y_2 \phi_2 P \quad (3.3)$$

where y_2 is the solubility (mole fraction) of the solid in the supercritical fluid phase, and ϕ_2 is the fugacity coefficient of the solid in the supercritical phase.

where y_2 is the solubility (mole fraction) of the solid in the supercritical fluid phase, and ϕ_2 is the fugacity coefficient of the solid in the supercritical phase.

From the condition of equal fugacities for the solid in both phases (Equation 3.1), the solubility of the solid in the fluid phase is

$$y_2 = P_2^s \frac{\exp[v_2^s(P - P_2^s)/RT]}{P\phi_2} \quad (3.4)$$

Experimental pure component vapor pressure values (P_2^s), and the solid volume v_2^s can be taken from the literature. ϕ_2 can be calculated from an EOS for a given set of pressure and temperature of the specified system.

3.2 Cubic Equations-of-State

Common cubic equations-of-state can be written in the form originally proposed by Schmidt and Wenzel (1980) for a generalized cubic EOS

$$P = \frac{RT}{v - b} - \frac{a}{v^2 + ubv + wb^2} \quad (3.5)$$

Parameters a and b are given by

$$a = a_c \alpha; \quad a_c = \Omega_a \frac{R^2 T_c^2}{P_c} \quad (3.6)$$

and

$$b = \Omega_b \frac{RT_c}{P_c} \quad (3.7)$$

The three cubic EOS employed in this work, along with their parameters, are summarized in Table 3.1

Table 3.1: Cubic Equations of State

EOS	n	w	Ω_a	Ω_b	α
RK	1	0	0.42748	0.08664	$\frac{1}{\sqrt{T_r}}$
SRK	1	0	0.42748	0.08664	$\left[1 + (0.480 + 1.574\omega - 0.176\omega^2)(1 - \sqrt{T_r})\right]^2$
PR	2	-1	0.45724	0.07780	$\left[1 + (0.37464 + 1.54226\omega - 0.26992\omega^2)(1 - \sqrt{T_r})\right]^2$

The composition dependence of parameters a and b for the mixture is represented by the conventional vdW mixing rules. For a binary mixture,

$$a = a_{11}y_1^2 + 2y_1y_2a_{12} + a_{22}y_2^2 \quad (3.8)$$

and

$$b = b_{11}y_1^2 + 2y_1y_2b_{12} + b_{22}y_2^2 \quad (3.9)$$

where y_i is the mole fraction of component i , and a_{ii} and b_{ii} are given by Equations 3.6 and 3.7, respectively. The combining rules for the cross parameters a_{12} and b_{12} are

$$a_{12} = \sqrt{a_{11}a_{22}}(1 - k_{12}) \quad (3.10)$$

and

$$b_{12} = \frac{b_{11} + b_{22}}{2}(1 - l_{12}) \quad (3.11)$$

where k_{12} and l_{12} are the binary interaction parameters. When l_{12} is set equal to zero, Equation 3.9 simplifies to

$$b = \sum_{i=1}^2 y_i b_i = y_1 b_1 + y_2 b_2 \quad (3.12)$$

A cube-root mixing rule for the parameter a was recently proposed by Pongsiri and Viswanath (1989) and corrected by Caballero and Estévez (1991). It is given by

$$a^{1/3} = \frac{y_1 (a_{11}M_1)^{1/3} + y_2 (a_{22}M_2)^{1/3}}{y_1(M_1)^{1/3} + y_2(M_2)^{1/3}} + (\sqrt{a_{11}a_{22}})^{1/3} y_1 y_2 (1 - k_{12}) \quad (3.13)$$

where M_i is the molecular mass of component i . The parameter b is calculated using the conventional vdW mixing and combining rules.

3.3 Fugacity Coefficient from Cubic EOS

The fugacity coefficient ϕ_i is given by (Prausnitz et al., 1986):

$$\ln \phi_i = \frac{1}{RT} \int_V^\infty \left[\left(\frac{\partial P}{\partial n_i} \right)_{T, V, n_{j \neq i}} - \frac{RT}{V} \right] dV - \ln z \quad (3.14)$$

where V is the total volume and $z = Pv/RT$ is the compressibility factor of the mixture, and n_i and n_j are the mole numbers of component i and j , respectively. Equations of state which are pressure-explicit can now be used to determine the analytical

form of $(\partial P/\partial n_i)_{T, V, n_j}$ for calculating the fugacity coefficient. However, since EOS mixing rules directly relate to mole fractions rather than to mole numbers, derivatives with respect to mole fractions would greatly facilitate the treatment of various mixing rules in cubic EOS for introducing functional modifications in the expression for component fugacity coefficients. Szarawara and Gawdzik (1988) developed a cohesive technique of generalizing the composition dependence of the component fugacity coefficients for the various types of two-parameter cubic EOS which represent $P = \mathcal{F}(T, v, a, b)$:

$$\ln \phi_i = \frac{Pv}{RT} - 1 + \frac{1}{RT} \int_v^\infty \mathcal{F} dv - \int_{v^\circ}^\infty \frac{dv}{v} + \frac{\left[\frac{\partial a}{\partial y_i} - \sum y_j \frac{\partial a}{\partial y_j} \right]}{RT} \int_v^\infty \frac{\partial \mathcal{F}}{\partial a} dv + \frac{\left[\frac{\partial b}{\partial y_i} - \sum y_j \frac{\partial b}{\partial y_j} \right]}{RT} \int_v^\infty \frac{\partial \mathcal{F}}{\partial b} dv \quad (3.15)$$

where v° is the molar volume of the ideal gas.

The generalized Schmidt-Wenzel EOS (Equation 3.5) with $\Delta > 0$, where $\Delta = u^2 - 4w$, represents the cubic EOS of Table 3.1 used in the present study. Equation 3.15 may be integrated in a general form when Equation 3.5 is used with $\Delta > 0$. The result is

$$\ln \phi_i = \left(\frac{\bar{b}_i - \sum y_j \bar{b}_j}{b} + 1 \right) (z - 1) - \ln(z - B) - \frac{A}{B\sqrt{\Delta}} \left[1 + \left(\frac{\bar{a}_i - \sum y_j \bar{a}_j}{a} \right) - \left(\frac{\bar{b}_i - \sum y_j \bar{b}_j}{b} \right) \right] \ln \left(\frac{z - r_2 B}{z - r_1 B} \right) \quad (3.16)$$

where

$$r_1 = \left(\frac{-u + \sqrt{\Delta}}{2} \right); \quad r_2 = \left(\frac{-u - \sqrt{\Delta}}{2} \right) \quad (3.17)$$

$$\bar{b}_i = \frac{\partial b}{\partial y_i}; \quad \sum y_j \bar{b}_j = \sum y_j \frac{\partial b}{\partial y_j} \quad (3.18)$$

$$\bar{a}_i = \frac{\partial a}{\partial y_i}; \quad \sum y_j \bar{a}_j = \sum y_j \frac{\partial a}{\partial y_j} \quad (3.19)$$

$$A = \frac{aP}{(RT)^2} \quad (3.20)$$

$$B = \frac{bP}{RT} \quad (3.21)$$

Further transformations of Equation 3.16 depend on the mixing rules.

Use of the quadratic vdW mixing rules

$$a = \sum_i \sum_j y_i y_j a_{ij} \quad (3.22)$$

and

$$b = \sum_i \sum_j y_i y_j b_{ij} \quad (3.23)$$

results in

$$\bar{a}_i = \frac{\partial a}{\partial y_i} = 2 \sum y_j a_{ij} \quad (3.24)$$

$$\sum y_j \bar{a}_j = \sum y_j \frac{\partial a}{\partial y_j} = 2a \quad (3.25)$$

$$\bar{b}_i = \frac{\partial b}{\partial y_i} = 2 \sum y_j b_{ij} \quad (3.26)$$

$$\sum y_j \bar{b}_j = \sum y_j \frac{\partial b}{\partial y_j} = 2b \quad (3.27)$$

In Equations 3.24 and 3.26, a_{ij} and b_{ij} are given by Equation 3.10 and 3.11 respectively.

If cube-root mixing rule for a is employed (Equation 3.13), we get

$$\bar{a}_i = \frac{\partial a}{\partial y_i} = 3a^{\frac{2}{3}} \left[\frac{M_j^{\frac{1}{3}} (a_{ii} M_i)^{\frac{1}{3}} - M_i^{\frac{1}{3}} (a_{jj} M_j)^{\frac{1}{3}}}{\left(\sum y_k M_k^{\frac{1}{3}} \right)^2} + (\sqrt{a_{11} a_{22}})^{\frac{1}{3}} (1 - k_{ij}) (1 - 2y_i) \right] \quad (3.28)$$

and

$$\begin{aligned} \sum y_j \bar{a}_j = \sum y_j \frac{\partial a}{\partial y_j} &= 3a^{\frac{2}{3}} \left[\frac{(y_i - y_j) \left\{ M_j^{\frac{1}{3}} (a_{ii} M_i)^{\frac{1}{3}} - M_i^{\frac{1}{3}} (a_{jj} M_j)^{\frac{1}{3}} \right\}}{\left(\sum y_k M_k^{\frac{1}{3}} \right)^2} \right] \\ &+ 3a^{\frac{2}{3}} \left[\left(\sqrt{a_{ii} a_{jj}} \right)^{\frac{1}{3}} (1 - k_{ij}) \left\{ 1 - 2(y_i^2 + y_j^2) \right\} \right] \quad (3.29) \end{aligned}$$

3.4 A Simplified Cubic EOS Model for Solubility

The simplified model presented below is especially advantageous for correlating solubility of high-molecular-weight (HMW) solids in supercritical fluids, since it does not require the critical properties of the solute (which are often not available). This model has the following key features (Estévez et al., 1994):

will be highly dependent on the method used to estimate the solute's critical properties).

- Since solubilities are usually low, infinite-dilution compressibility factor is used to compute the infinite-dilution fugacity coefficient of the solute in the supercritical fluid phase.

The first feature renders the use of critical property estimation methods needless. The second feature tremendously simplifies the computational aspect of the model, since it eliminates the need for an iterative scheme. Using the infinite-dilution concept renders the compressibility factor and the fugacity coefficient of the solute in the supercritical fluid-phase independent of the solubility of the solute. The conventional iterative procedure is thus substituted by a once-through calculation.

The modifications or simplifications herein can be used with any EOS. In this work, the RK, SRK and PR EOS have been used to illustrate the method. For these EOS, Equation 3.16 gives

$$\ln \phi_2 = \frac{b_2}{b}(z-1) - \ln(z-B) - \frac{A}{B\sqrt{\Delta}} \left[\frac{2(y_1 a_{12} + y_2 a_2)}{a} - \frac{b_2}{b} \right] \ln \left[\frac{z - r_2 B}{z - r_1 B} \right] \quad (3.30)$$

As can be seen, y_2 appears explicitly and implicitly through z which is a function of composition in the supercritical phase. For the fluid parameters, a and b , standard vdW mixing rules are used (Equations 3.8 and 3.12).

For the infinite dilution limit ($y_2 = 0, y_1 = 1, a = a_1$ and $b = b_1$), Equation 3.30 reduces to

$$\ln \phi_2^\infty = \frac{b_2}{b}(z^\infty - 1) - \ln(z^\infty - B) - \frac{A}{B\sqrt{\Delta}} \left[\frac{2a_{12}}{a_1} - \frac{b_2}{b_1} \right] \ln \left[\frac{z^\infty - r_2 B}{z^\infty - r_1 B} \right] \quad (3.31)$$

Using the standard combination rule for a_{12} (Equation 3.30 with $k_{12} = 0$), Equation 3.31 can be written as

$$\ln \phi_2^\infty = \frac{b_2}{b}(z^\infty - 1) - \ln(z^\infty - B) - \frac{A}{B\sqrt{\Delta}} \left[2\sqrt{\frac{a_2}{a_1}} - \frac{b_2}{b_1} \right] \ln \left[\frac{z^\infty - r_2 B}{z^\infty - r_1 B} \right] \quad (3.32)$$

Defining parameters α and β as

$$\alpha = \sqrt{\frac{a_2}{a_1}} \quad (3.33)$$

$$\beta = \frac{b_2}{b_1} \quad (3.34)$$

Defining parameters α and β as

$$\alpha = \sqrt{\frac{a_2}{a_1}} \quad (3.33)$$

$$\beta = \frac{b_2}{b_1} \quad (3.34)$$

the final equation for the infinite-dilution fugacity coefficient of the solute in the supercritical fluid phase is given by

$$\ln \phi_2^\infty = \beta(z^\infty - 1) - \ln(z^\infty - B) - \frac{A(2\alpha - \beta)}{B\sqrt{\Delta}} \ln \left[\frac{z^\infty - r_2 B}{z^\infty - r_1 B} \right] \quad (3.35)$$

In all the equations for $\ln \phi_2^\infty$ (Equations 3.31, 3.32, 3.35), A and B are defined as infinite dilution limits of Equations 3.20 and 3.21, respectively. Reduced properties are normalized with respect to the solvent critical properties.

Chapter 4

Results and Discussion

4.1 Physical Properties

The physical properties (M , T_c , P_c and ω) of the supercritical solvents are presented in Table 4.1. The solid solutes used in this study are presented in Table 4.2, along with their critical properties (T_c , P_c), molecular weights, acentric factors and molar volumes. The constants of the vapor pressure relation for the solid solutes are also presented in Table 4.2. Table 4.3 presents the data for the sublimation pressures of PCBs at the temperatures of interest. The references from which the data were taken are also shown in the tables.

Table 4.1: Physical Properties of the Supercritical Solvents

Solvent	M.W.	T_c (K)	P_c (bar)	ω	References
CO ₂	44.010	73.8	304.1	0.239	Reid et al., 1987
C ₂ H ₆	30.070	48.8	305.4	0.099	Reid et al., 1987
C ₂ H ₄	28.054	50.4	282.4	0.089	Reid et al., 1987
CClF ₃	104.459	38.7	302.0	0.198	Reid et al., 1987
CHF ₃	70.013	48.6	299.3	0.260	Reid et al., 1987

Table 4.2: Physical Properties of the Solid Solutes

Solute	M.W.	T _c (K)	P _c (bar)	ω	v^s (cc/mol)	$P^{sat}(\text{bar}) = 10^{\left(A - \frac{B}{T(K) - C}\right)}$			References
						A	B	C	
Acridine	179.222	NA	NA	NA	178.0	13.721	4740.0	0.0 ¹	Schmitt & Reid, 1986
Anthracene	178.234	869.3	21.9	0.370	142.0	26.805	11402.0	0.0 ^a	Barna et al., 1995
Benzoic Acid	122.124	752.0	45.6	0.62	96.50	10.5432	4190.7	-125.2 ^b	Reid et al., 1987
4,4'-DCB	223.101	847.5	30.5	0.453	155.0	NA	NA	NA	Yu et al., 1995
Diphenyl	154.212	789.0	38.5	0.372	131.0	9.4068	4262.0	0.0	Barna et al., 1995
2,3-Dimethylnaphthalene	156.228	785.0	32.2	0.424	154.7	9.0646	4302.5	0.0	Barna et al., 1995
2,6-Dimethylnaphthalene	156.228	777.0	32.2	0.420	139.23	9.4286	4419.5	0.0	Barna et al., 1995
Fluorene	165.216	821.0	29.9	0.407	136.8	9.2050	4561.8	0.0	Barna et al., 1995
2,2',3,3',4,4'-HCB	360.903	889.0	24.5	0.716	204.0	NA	NA	NA	Yu et al., 1995
Hexamethylbenzene	162.276	752.0	23.8	0.498	152.37	8.1340	3855.0	21.0	Barna et al., 1995
Naphthalene	128.174	748.4	40.5	0.302	110.30	8.5830	3733.9	0.0	Barna et al., 1995
1,4-Naphthoquinone	158.156	NA	NA	NA	111.0	14.735	4739.4	0.0 ¹	Schmitt & Reid, 1986
Phenanthrene	178.234	882.6	31.7	0.437	151.0	9.6310	4873.4	0.0	Barna et al., 1995
Pyrene	202.256	936.0	26.0	0.494	158.5	8.3946	4904.0	0.0	Barna et al., 1995
2,3',4',5-TCB	292.002	878.0	27.2	0.562	180.0	NA	NA	NA	Yu et al., 1995
Triphenylmethane	100.205	863.0	22.4	0.576	214.3	9.7858	5228.0	0.0	Barna et al., 1995

$$^a P^{sat}(\text{bar}) = 1.3332 \times 10^{-3} \exp\left(A - \frac{B}{T(K) - C}\right)$$

$$^b P^{sat}(\text{bar}) = \exp\left(A - \frac{B}{T(K) - C}\right), \text{Ref: Schmitt \& Reid, 1986}$$

$$^c P^{sat}(\text{bar}) = 10^{-5} 10^{\left(A - \frac{B}{T(K) - C}\right)}$$

Table 4.3: Sublimation Pressures of Polychlorinated Biphenyls

Solute	$P^{sat}(\text{bar})$			References
	308.1 K	313.1 K	323.1 K	
4,4'-DCB	NA	1.986	6.594	Yu et al., 1995
2,2',3,3',4,4'-HCB	NA	0.2845	1.0104	Yu et al., 1995
2,3',4',5-TCB	0.2359	0.4546	1.5886	Yu et al., 1995

4.2 Data Reduction and Parameter Estimation Criteria

In the rigorous EOS modeling of the solubilities, the binary interaction parameters k_{ij} and l_{ij} which account for the solute-solvent energetic interactions and size and shape differences, respectively were determined by a regression method minimizing an objective function which measures the deviations between experimental ($y_{2,exp}$) and calculated ($y_{2,cal}$) values of the solute solubility. The objective function (OF) used is the root-mean-square (rms) of the absolute deviations, which is defined as

$$OF = \sqrt{\frac{\sum_{i=1}^N (y_{2,exp} - y_{2,cal})^2}{N}} \quad (4.1)$$

where N is the number of experimental data points. Rosenbrock's optimization algorithm (Beveridge & Schechter, 1970) is used to minimize OF. The quality of the fitting of the EOS to the experimental solubility data is measured in terms of a percentage average error, AE (%), defined as

$$\begin{aligned} AE(\%) &= \frac{\sqrt{\frac{\sum_{i=1}^N (y_{2,exp} - y_{2,cal})^2}{N}}}{\bar{y}_{2,exp}} \times 100 \\ &= \frac{OF}{\bar{y}_{2,exp}} \times 100 = N \frac{OF}{\sum_{i=1}^N y_{2,exp}} \times 100 \end{aligned} \quad (4.2)$$

The listing of the computer program used to obtain the optimum values of the BIPs appears in Appendix A.

The parameters α and β of the simplified model were determined by minimizing an objective function which measures the relative deviations between experimental

and calculated values of the solubility. The objective function (OF) used is:

$$\text{OF} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{y_{2,\text{exp}} - y_{2,\text{cal}}}{y_{2,\text{exp}}} \right)^2} \quad (4.3)$$

The listing of the computer program used to obtain the optimum values of the parameters α and β appears in Appendix B.

4.3 Results and Discussion

Correlation results for binary systems of 14 solutes and 5 solvents at different temperatures for each of the three EOS tested are given in Tables 4.4 through 4.7. Tables 4.4 and 4.5 show the optimum k_{12} values and percent average errors when the EOS mixture cohesive energy parameter a is expressed by the quadratic (QMR) and the cube-root (CMR) mixing rule, respectively; the mixture repulsive energy parameter, b , of the EOS being expressed by the linear mixing rule. Our k_{12} values for the PR-EOS using quadratic mixing rules are in very good agreement with the values reported by other researchers. Most recently Yu et al. (1995) reported k_{12} values of 0.0605 and 0.0582 for the system 4,4'-DCB/CO₂ at 313.1 K and 323.1 K, respectively. Our corresponding values are 0.0604 and 0.0571. Bartle et al. (1991) reported a k_{12} value of 0.090 for the system fluorene/CO₂ at 308.15 K using a simplified form of the PR-EOS for dilute solutions; our calculated value for the same system using the conventional PR-EOS is 0.0826.

Table 4.6 and 4.7 summarize the optimum k_{12} and l_{12} values and percent average errors when parameter a of the EOS is expressed by the QMR and CMR, respectively and the parameter b is expressed by the quadratic mixing rule. The binary interaction parameter k_{12} measures the deviation from geometric mean intermolecular attractions assumed for the unlike cohesive energy parameter a_{12} , while the binary interaction parameter l_{12} measures the deviation from arithmetic mean intermolecular repulsions assumed for the unlike repulsive energy parameter b_{12} . Both k_{12} and l_{12} , being purely empirical parameters, can be either positive or negative. A physical interpretation has been ascribed to negative k_{12} values as being indicative of the presence in the mixture of specific chemical interactions such as hydrogen bonding (McHugh and Krukoni, 1982). However, not much significance can be given to such an interpretation for a

parameter which is essentially empirical. The values of the BIPs (k_{12} and l_{12}) depend on the accuracy of the physical properties (T_c , P_c and ω) used to evaluate the pure component parameters of the EOS, the source of the experimental solubility data, and the form of the objective function used to minimize the deviations between calculated and experimental values of solubility. No definite trends can be observed in the behaviour of the k_{12} and l_{12} values with increasing temperature. For any particular system, however, the behaviour of the l_{12} values mirrors that of the k_{12} values in general, that is, if k_{12} increases with temperature, so does l_{12} ; if k_{12} decreases, l_{12} also decreases.

The results of correlation shown in Table 4.6 and 4.7 can be considered satisfactory, considering the uncertainty in experimental measurements of low solubility values, e.g. 10-20 % below 10^{-4} mole fraction and increasing to 40 % for the lower end of the solubility data (below 10^{-5} mole fraction) as reported by Johnston et al. (1982), as well as the uncertainty of the T_c , P_c and ω values of solute. Comparison of the overall percent average errors of Table 4.4 and 4.5 with those of Table 4.6 and 4.7 shows that, as expected, the introduction of the second binary interaction parameter, l_{12} , greatly improves the accuracy of the calculated solubility values. The calculated solubilities of anthracene in supercritical carbon dioxide at 323.15 K using the PR EOS/QMR formulation are shown in Figure 4.1. The plots clearly demonstrate that using the two BIPs (k_{12} and l_{12}) results in a marked improvement in the fit of calculated solubilities with the experimental data in comparison to that resulting from the use of k_{12} alone or setting $k_{12} = 0$.

From the overall percent average errors of Tables 4.4 to 4.7, we can conclude that the cube-root mixing rule performs as well as the conventional vdW mixing rule in representing solid- supercritical fluid equilibria and, in general no distinction can be made among the EOS. However, on closer inspection, differences among the EOS are found when the errors for individual systems are compared. Some typical results are shown in Figures 4.2- 4.5. The calculated solubilities of benzoic acid in fluoroform at 328.25 K using different EOS and a single BIP (k_{12}) in the cube-root mixing rule are shown in Figure 4.2. Figure 4.3 presents the results for the system naphthalene/fluoroform at 328.15 K using different EOS/CMR formulation with optimum k_{12} and l_{12} values . Figures 4.4 and 4.5 present the results for the systems fluorene/ethylene at 343.15 K and benzoic acid/ CO_2 at 338 K, respectively

using different EOS with quadratic mixing rule; Figure 4.4 is based on the use of optimum k_{12} alone whereas optimum k_{12} and l_{12} have been used in generating the plots in Figure 4.5. For the systems represented in Figures 4.2 and 4.3, the data are best correlated by the SRK EOS; on the other hand, for the systems represented in Figures 4.4 and 4.5, the data appear to be better correlated by the PR EOS than by the RK or SRK EOS. This observation could indicate that, besides the use of BIPs, the functional form of the attractive term of the EOS plays an important role in the ability of the equation to account for the effect of nondispersive-type forces in solid-supercritical fluid systems. In systems where dispersive forces predominate, no significant difference is likely to be found in the ability of the equations to model solid-supercritical fluid equilibria. When specific chemical forces are present, however, the nature and strength of these forces, as determined by the type of solvent, will influence the ability of these equations to model the system. This ability will depend on how well the functional form of the attractive term of the EOS can account for the effect of these chemical forces.

For some systems the nature of the supercritical solvent is observed to have a marked influence on correlation results (Tables 4.6 and 4.7). For example, in benzoic acid/solvent systems a considerable increase in percent average error is noticed when the supercritical solvent is changed from CO_2 to C_2H_4 . A possible explanation for this observation could be related to the differences in the polarizabilities of the two solvents and the effect of this difference on the solute-solvent interaction constant a_{12} . The polarizability of C_2H_4 is $42.52 \times 10^{-25} \text{ cm}^3$, while that of CO_2 is $29.11 \times 10^{-25} \text{ cm}^3$ (Lide, 1990); the ratio of the polarizability of C_2H_4 to that of CO_2 being approximately 1.5. For the same solute, and upon changing the solvent from CO_2 to C_2H_4 at the same temperature, it is therefore to be expected that the unlike interaction constant a_{12} should be considerably larger for C_2H_4 than for CO_2 . However a much lower ratio of a_{12} in C_2H_4 to that in CO_2 is predicted using critical properties. The increase in percent average error may point at the inadequacy of using critical properties to predict successfully the interaction constant in mixtures that are highly asymmetric.

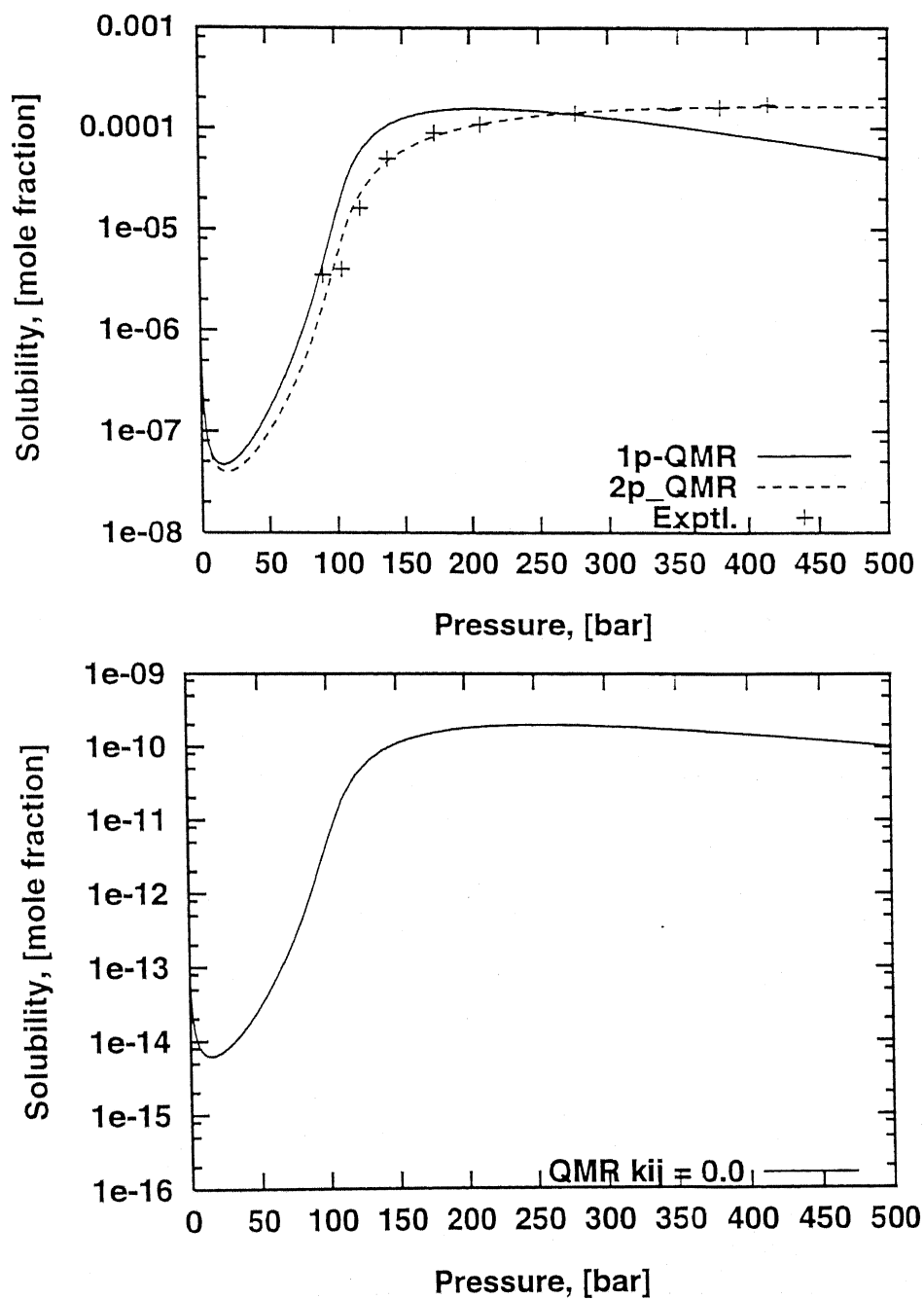


Figure 4.1: Calculated Solubilities of Anthracene in Supercritical CO₂ at 323.15 K Using the PR EOS/QMR formulation: Role of Binary Interaction Parameters

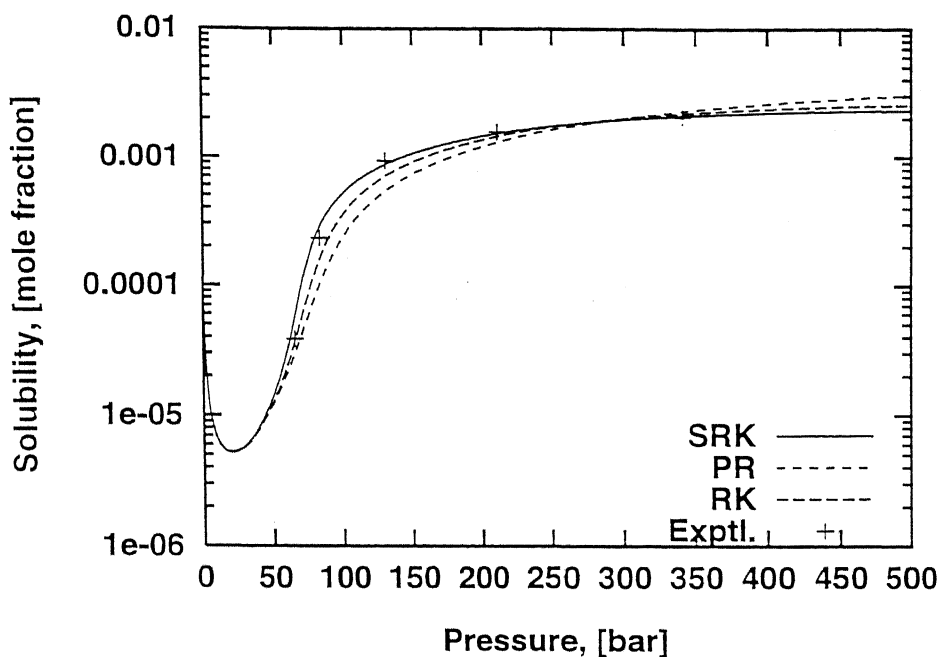


Figure 4.2: Calculated Solubilities of Benzoic Acid in Fluoroform at 328.25 K Using Different EOS/CMR Formulation with Optimum k_{12}

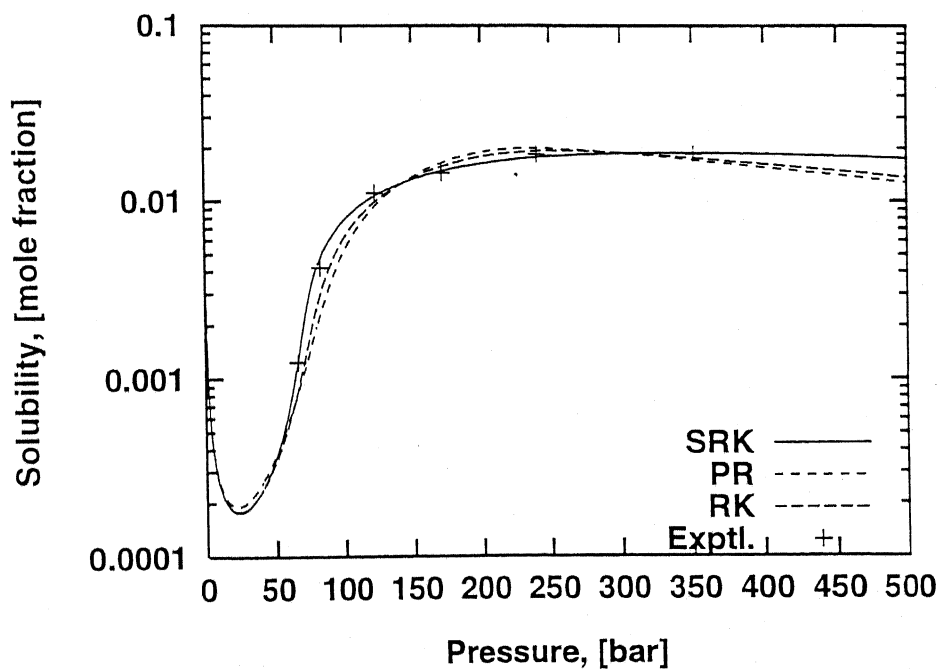


Figure 4.3: Calculated Solubilities of Naphthalene in Fluoroform at 328.25 K Using Different EOS/CMR Formulation with Optimum k_{12} & l_{12}

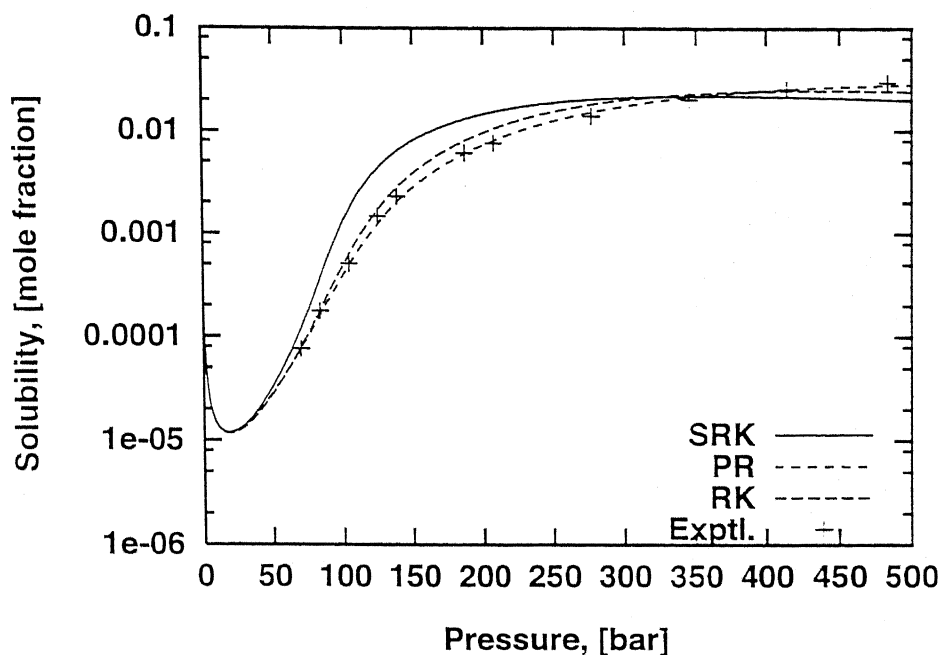


Figure 4.4: Calculated Solubilities of Fluorene in Ethylene at 343.15 K Using Different EOS/QMR Formulation with Optimum k_{12}

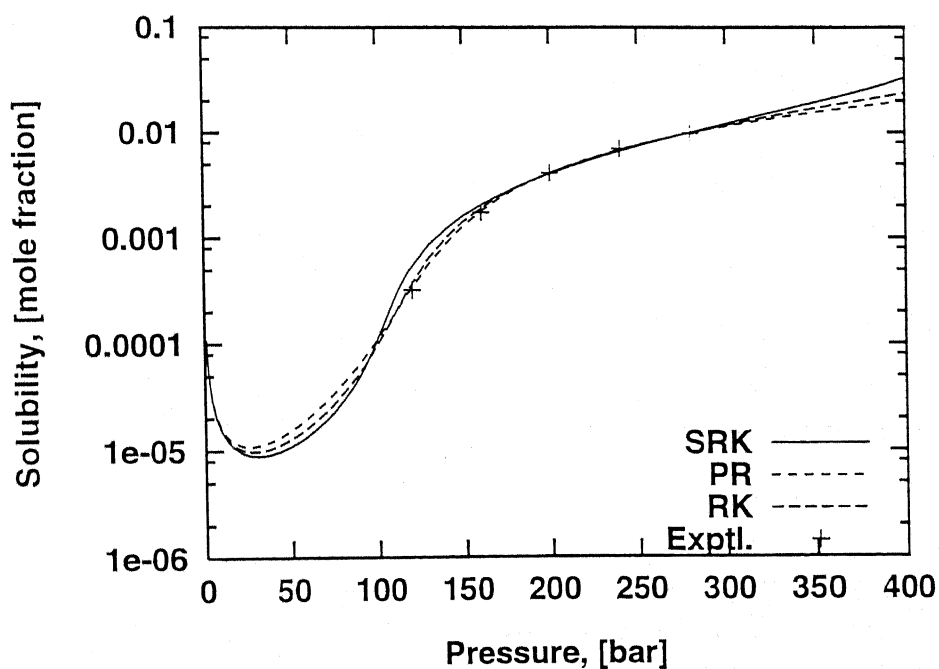


Figure 4.5: Calculated Solubilities of Benzoic Acid in CO_2 at 338 K Using Different EOS/QMR Formulation with Optimum k_{12} & l_{12}

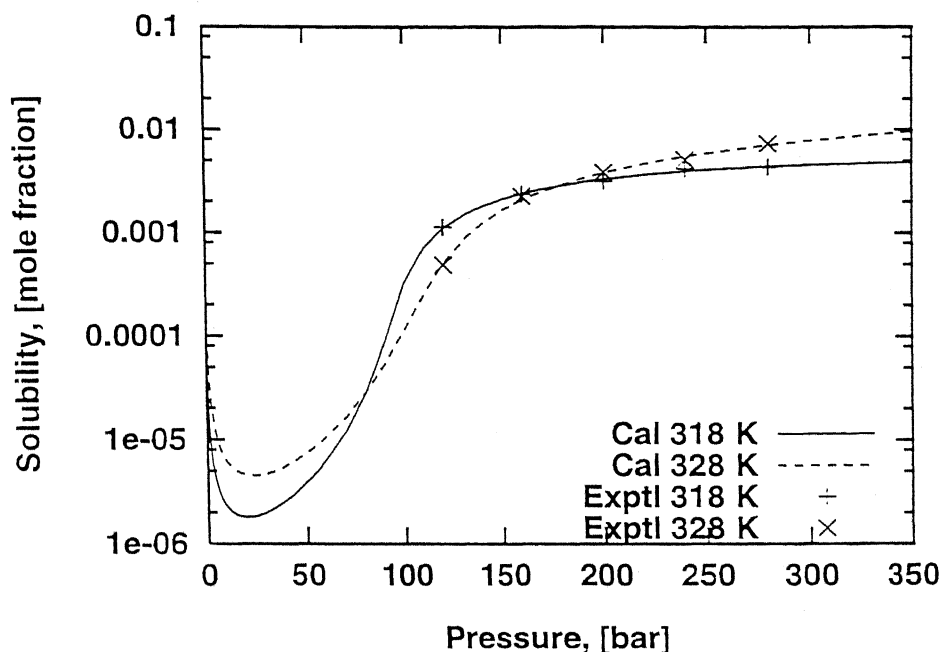


Figure 4.6: Calculated Solubilities of Benzoic Acid in Supercritical CO_2 at Different Temperatures Using the Simplified Model with the PR EOS

Due to nonavailability of the critical properties for the solutes acridine and 1,4-naphthoquinone, the rigorous EOS modeling was not done for systems made up of these solutes. Table 4.8 presents optimum solute-to-solvent parameter ratios α and β and percent average errors for binary systems of 15 solutes (including acridine and 1,4-naphthoquinone) and 5 solvents at different temperature for each of three EOS using the simplified method. In general, both parameters decrease with increasing temperatures. The rate of change of β with temperature is marginally higher (in absolute value) than the corresponding value for α . Comparison of overall percent average errors shows that no distinction can be made among the EOS. To illustrate typical prediction capabilities of the simplified model, four systems have been randomly chosen, and their solubility curves at different temperatures calculated using the PR EOS are shown in Figures 4.6 through 4.9. Also included in these Figures are the experimental data. Qualitative observation of the curves indicate that in all cases the simplified model gives fairly accurate predictions of solubility.

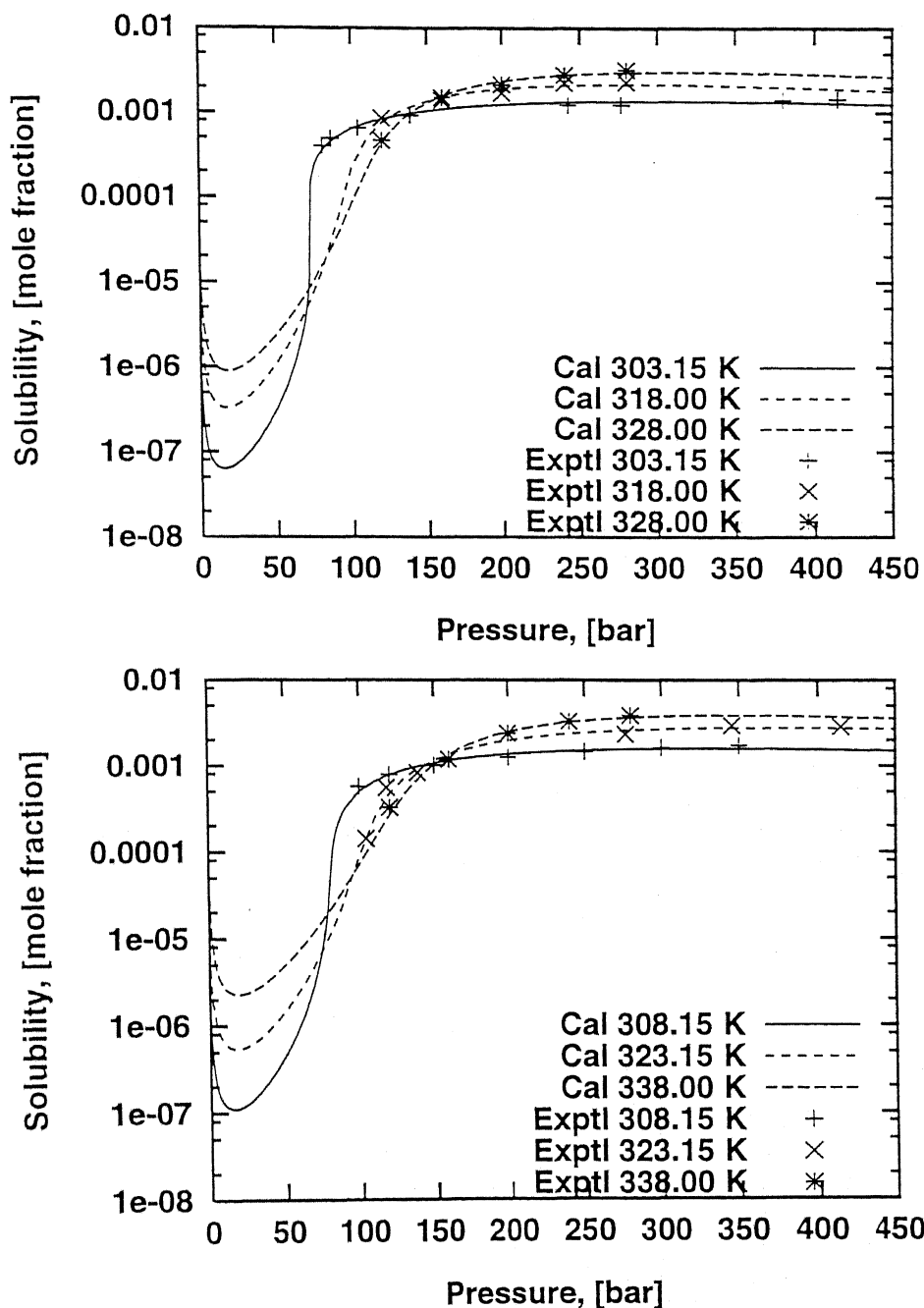


Figure 4.7: Calculated Solubilities of Phenanthrene in Supercritical CO₂ at Different Temperatures Using the Simplified Model with the PR EOS

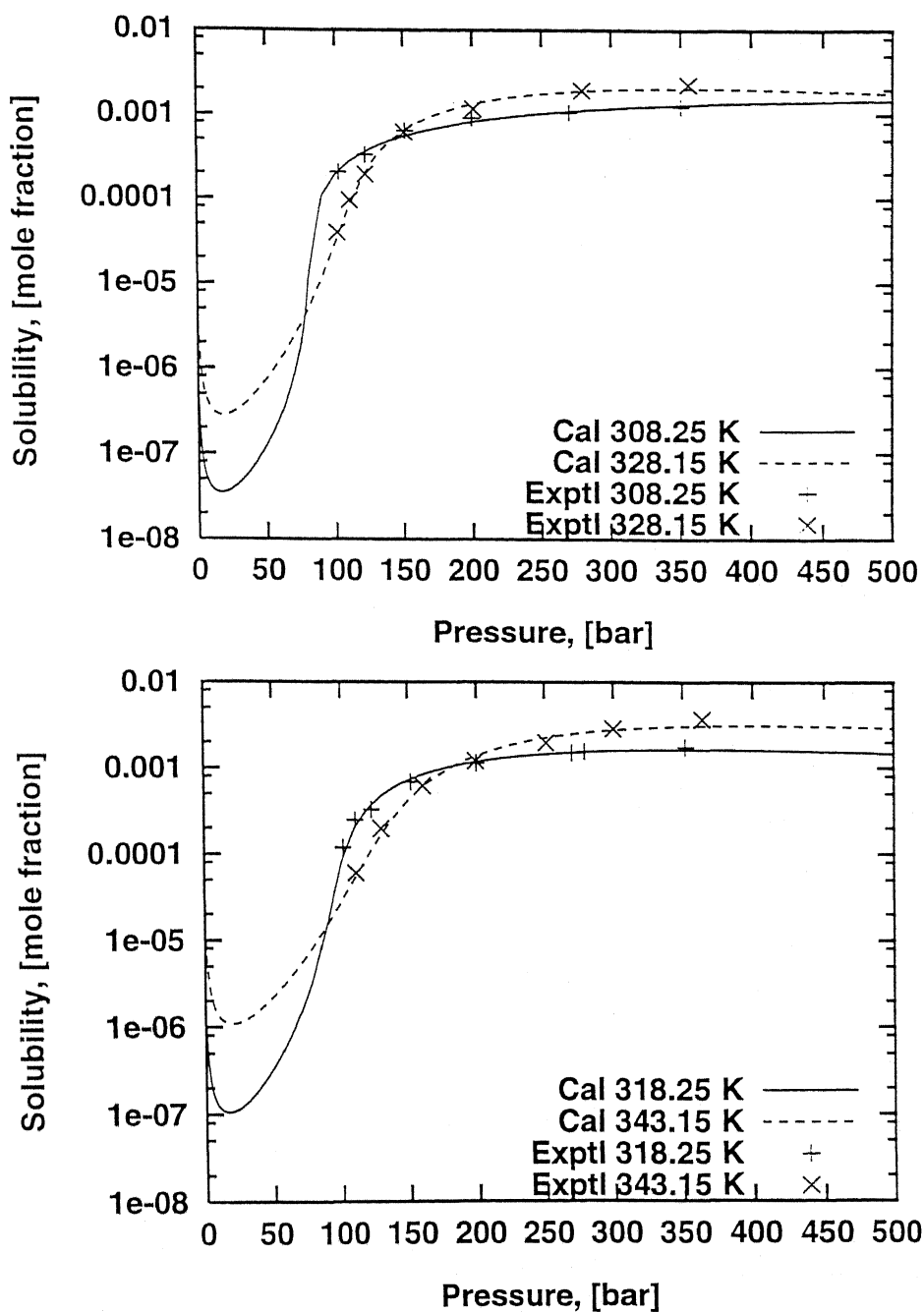


Figure 4.8: Calculated Solubilities of Acridine in Supercritical CO₂ at Different Temperatures Using the Simplified Model with the PR EOS

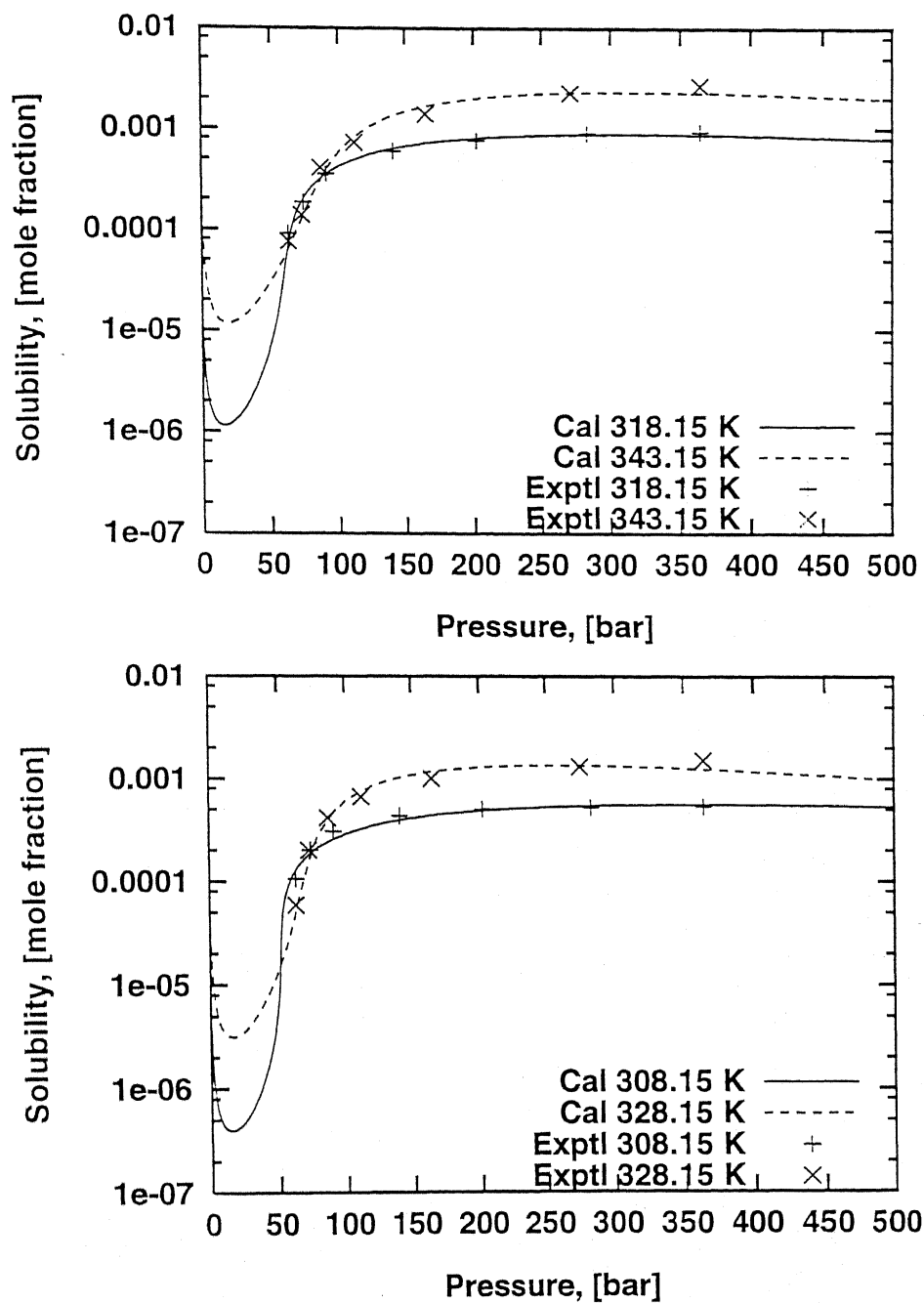


Figure 4.9: Calculated Solubilities of 1,4-Naphthaquinone in Supercritical Ethane at Different Temperatures Using the Simplified Model with the PR EOS

Table 4.4: EOS Modeling of Solubility: Optimum k_{12} Using the Quadratic Mixing Rule (QMR)

Equation of State			S-R-K		P-R		R-K		References
Solute	Solvent	Temp.	N	k_{ij}	AE(%)	k_{ij}	AE(%)	k_{ij}	
Anthracene	CHF ₃	328.15	3	-0.1846	79.80	0.1209	40.35	0.0510	Schmitt and Reid (1986)
Anthracene	CHF ₃	343.15	3	-0.1739	73.61	0.1138	28.33	0.0440	Schmitt and Reid (1986)
Anthracene	CO ₂	303.15	4	-0.2670	74.14	0.0977	68.68	0.0269	Johnston et al. (1982)
Anthracene	CO ₂	308.00	5	-0.2632	57.51	0.0964	48.37	0.0236	Kosal and Holder (1987)
Anthracene	CO ₂	318.00	6	-0.2472	52.21	0.0968	28.17	0.0235	Kosal and Holder (1987)
Anthracene	CO ₂	323.15	10	-0.2445	90.54	0.0867	57.95	0.0148	Johnston and Eckert (1981)
Anthracene	CO ₂	343.15	9	-0.2064	102.97	0.0830	48.32	0.0136	Johnston and Eckert (1981)
Anthracene	C ₂ H ₆	303.15	6	-0.3445	63.05	0.0539	54.34	-0.0323	Johnston et al. (1982)
Anthracene	C ₂ H ₆	308.15	4	-0.3281	68.01	0.0591	57.92	-0.0263	Johnston et al. (1982)
Anthracene	C ₂ H ₆	323.15	10	-0.3150	58.72	0.0538	44.21	-0.0353	Johnston et al. (1982)
Anthracene	C ₂ H ₆	343.15	7	-0.2890	54.69	0.0506	35.49	-0.0398	Johnston et al. (1982)
Anthracene	C ₂ H ₄	323.15	7	-0.3323	66.10	0.0305	40.36	-0.0621	Johnston and Eckert (1981)
Anthracene	C ₂ H ₄	343.15	9	-0.2814	58.91	0.0428	29.00	-0.0479	Johnston and Eckert (1981)
Anthracene	C ₂ H ₄	358.15	13	-0.2676	47.27	0.0374	22.40	-0.0533	Johnston and Eckert (1981)
Benzoic Acid	CClF ₃	318.15	6	-0.3586	6.22	0.0743	11.58	-0.1076	Schmitt and Reid (1986)
Benzoic Acid	CClF ₃	328.15	6	-0.3347	13.10	0.0692	8.09	-0.1055	Schmitt and Reid (1986)
Benzoic Acid	CHF ₃	318.25	5	-0.3681	5.66	0.0627	20.78	-0.1166	Schmitt and Reid (1986)
Benzoic Acid	CHF ₃	328.25	5	-0.3456	5.59	0.0595	20.04	-0.1137	Schmitt and Reid (1986)

continued on next page

continued from previous page

Equation of State				S-R-K		P-R		R-K		References
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)	
Benzoic Acid	CO ₂	318.00	5	-0.3974	5.14	0.0412	10.49	-0.1381	5.79	Kurnik et al. (1981)
Benzoic Acid	CO ₂	328.00	5	-0.3805	18.71	0.0306	5.07	-0.1424	9.67	Kurnik et al. (1981)
Benzoic Acid	CO ₂	338.00	5	-0.3621	26.92	0.0203	3.57	-0.1451	13.68	Kurnik et al. (1981)
Benzoic Acid	C ₂ H ₄	318.00	5	-0.4714	74.73	-0.0203	80.48	-0.2182	79.06	Schmitt and Reid (1986)
Benzoic Acid	C ₂ H ₄	328.00	5	-0.4404	58.24	-0.0195	64.83	-0.2124	63.34	Schmitt and Reid (1986)
Benzoic Acid	C ₂ H ₄	338.00	5	-0.4214	42.58	-0.0275	49.23	-0.2165	47.78	Schmitt and Reid (1986)
Benzoic Acid	C ₂ H ₆	308.15	8	-0.4477	9.25	0.0347	13.19	-0.1603	9.76	Schmitt and Reid (1986)
Benzoic Acid	C ₂ H ₆	318.15	8	-0.4170	8.04	0.0394	14.66	-0.1520	9.82	Kurnik et al. (1981)
Benzoic Acid	C ₂ H ₆	328.15	7	-0.4113	10.16	0.0273	6.26	-0.1636	4.59	Kurnik et al. (1981)
Benzoic Acid	C ₂ H ₆	343.15	7	-0.3924	13.88	0.0167	6.83	-0.1713	2.91	Kurnik et al. (1981)
Biphenyl	CO ₂	308.15	7	-0.3238	52.43	0.0443	41.85	-0.0472	46.47	Chung and Shing (1992)
Biphenyl	CO ₂	308.95	8	-0.2842	11.17	0.0735	18.46	-0.0172	12.92	McHugh and Paulaitis (1980)
Biphenyl	CO ₂	318.55	8	-0.2809	1.27	0.0658	10.86	-0.0250	3.35	McHugh and Paulaitis (1980)
Biphenyl	CO ₂	322.65	8	-0.2838	3.92	0.0619	9.26	-0.0305	2.09	McHugh and Paulaitis (1980)
Biphenyl	CO ₂	328.15	6	-0.2675	21.45	0.0576	8.94	-0.0279	9.04	Chung and Shing (1992)
Biphenyl	C ₂ H ₆	308.35	5	-0.3750	17.26	0.0136	21.75	-0.0874	18.69	Schmitt and Reid (1986)
Biphenyl	C ₂ H ₆	318.15	5	-0.3669	8.25	0.0114	18.75	-0.0907	14.73	Schmitt and Reid (1986)
4,4'-DCB	CO ₂	313.10	9	-0.3522	12.51	0.0604	14.65	-0.0585	12.27	Yu et al. (1995)
4,4'-DCB	CO ₂	323.10	8	-0.3396	11.50	0.0571	15.58	-0.0614	5.58	Yu et al. (1995)
2,3-DMN	CO ₂	308.00	5	-0.2770	5.10	0.0921	3.72	-0.0122	3.44	Kurnik et al. (1981)

continued on next page

continued from previous page

Equation of State									
Solute	Solvent	Temp.	N	S-R-K		P-R		R-K	
				k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)
2,3-DMN	CO ₂	318.00	5	-0.2547	6.41	0.0953	11.22	-0.0067	3.83
2,3-DMN	CO ₂	328.00	5	-0.2345	10.39	0.0962	10.29	-0.0027	1.09
2,3-DMN	C ₂ H ₄	308.00	6	-0.3740	15.28	0.0156	3.27	-0.1033	5.17
2,3-DMN	C ₂ H ₄	318.00	6	-0.3655	23.73	0.0087	1.79	-0.1105	8.84
2,3-DMN	C ₂ H ₄	328.00	6	-0.3694	23.66	0.0006	3.83	-0.1210	4.15
2,6-DMN	CO ₂	308.00	5	-0.2733	6.66	0.0911	5.55	-0.0119	6.10
2,6-DMN	CO ₂	318.00	5	-0.2630	14.17	0.0852	4.10	-0.0164	6.43
2,6-DMN	CO ₂	328.00	5	-0.2463	15.36	0.0833	8.15	-0.0154	7.77
2,6-DMN	C ₂ H ₄	308.00	6	-0.3761	16.05	0.0093	2.01	-0.1088	6.47
2,6-DMN	C ₂ H ₄	318.00	6	-0.3661	24.67	0.0022	4.47	-0.1159	10.89
2,6-DMN	C ₂ H ₄	328.00	6	-0.3582	29.95	-0.0047	1.87	-0.1227	10.58
Fluorene	CO ₂	303.15	7	-0.3024	25.89	0.0835	17.29	-0.0147	26.33
Fluorene	CO ₂	308.15	6	-0.2961	24.41	0.0826	12.57	-0.0164	21.72
Fluorene	CO ₂	323.15	9	-0.2728	33.76	0.0810	7.35	-0.0164	21.76
Fluorene	CO ₂	343.15	8	-0.2564	54.54	0.0724	8.12	-0.0262	29.74
Fluorene	C ₂ H ₄	298.15	6	-0.4107	27.65	0.0119	11.73	-0.1039	21.40
Fluorene	C ₂ H ₄	318.15	7	-0.3857	41.23	0.0058	14.78	-0.1122	27.90
Fluorene	C ₂ H ₄	343.15	11	-0.3616	51.78	-0.0046	7.85	-0.1232	24.82
2,2',3,3',4,4'-HCB	CO ₂	313.10	5	-0.3006	45.07	0.1829	35.14	-0.0035	43.68
2,2',3,3',4,4'-HCB	CO ₂	323.10	5	-0.2822	52.02	0.1822	32.92	-0.0013	45.28

continued on next page

continued on next page

continued from previous page										
Equation of State			S-R-K			P-R		R-K		References
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)	
HMB	CO ₂	303.15	7	-0.0474	56.24	0.2600	51.54	0.1655	56.53	Johnston et al. (1982)
HMB	CO ₂	323.15	10	-0.0047	92.22	0.2672	68.60	0.1772	80.97	Johnston et al. (1982)
HMB	CO ₂	343.15	10	0.0124	106.29	0.2532	72.98	0.1649	91.14	Johnston et al. (1982)
HMB	C ₂ H ₄	298.15	9	-0.1365	86.44	0.2009	65.25	0.0882	77.36	Johnston et al. (1982)
HMB	C ₂ H ₄	318.15	9	-0.1124	100.45	0.1949	71.38	0.0827	85.98	Johnston et al. (1982)
HMB	C ₂ H ₄	343.15	6	-0.0693	112.85	0.1928	67.76	0.0817	83.81	Johnston et al. (1982)
Naphthalene	CClF ₃	308.15	7	-0.2274	6.64	0.0833	13.67	0.0137	9.05	Schmitt and Reid (1986)
Naphthalene	CClF ₃	318.15	7	-0.2106	4.96	0.0833	15.50	0.0146	8.63	Schmitt and Reid (1986)
Naphthalene	CClF ₃	328.15	7	-0.1991	6.21	0.0798	14.29	0.0118	5.22	Schmitt and Reid (1986)
Naphthalene	CHF ₃	308.15	6	-0.2142	6.83	0.0887	19.77	0.0230	12.06	Schmitt and Reid (1986)
Naphthalene	CHF ₃	318.15	6	-0.2005	4.21	0.0857	19.82	0.0214	8.93	Schmitt and Reid (1986)
Naphthalene	CHF ₃	328.15	6	-0.1865	8.52	0.0830	17.66	0.0211	6.11	Schmitt and Reid (1986)
Naphthalene	CO ₂	308.15	9	-0.2211	2.46	0.0816	5.99	0.0203	3.54	McHugh and Paulaitis (1980)
Naphthalene	CO ₂	328.15	16	-0.2131	14.69	0.0642	7.56	0.0045	4.49	McHugh and Paulaitis (1980)
Naphthalene	C ₂ H ₆	293.15	7	-0.3170	24.49	0.0359	17.88	-0.0329	18.12	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	298.15	7	-0.3051	9.74	0.0377	8.41	-0.0325	7.23	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	308.15	6	-0.3028	11.91	0.0261	15.13	-0.0442	13.29	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	318.15	8	-0.2877	7.81	0.0273	17.40	-0.0442	13.38	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	328.15	13	-0.2923	2.61	0.0270	7.93	-0.0474	6.05	Johnston et al. (1982)
Naphthalene	C ₂ H ₄	298.15	10	-0.3269	4.60	-0.0009	16.36	-0.0690	13.10	Johnston and Eckert (1981)

continued on next page

continued on next page

continued from previous page									
Equation of State				S-R-K		P-R		R-K	
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)
Phenanthrene	C ₂ H ₄	343.15	8	-0.3177	31.57	0.0290	4.63	-0.0849	9.75
Pyrene	CO ₂	308.15	7	-0.2945	68.16	0.1326	53.01	0.0167	65.47
Pyrene	CO ₂	323.15	7	-0.2823	68.57	0.1254	36.62	0.0071	54.83
Pyrene	CO ₂	343.15	8	-0.2586	86.09	0.1173	28.49	-0.0004	58.75
Pyrene	C ₂ H ₄	313.15	8	-0.3963	50.20	0.0547	21.69	-0.0862	35.18
Pyrene	C ₂ H ₄	348.15	7	-0.3641	50.19	0.0431	16.66	-0.0997	29.96
2,3',4',5-TCB	CO ₂	308.10	6	-0.3727	12.87	0.0953	7.83	-0.0582	11.69
2,3',4',5-TCB	CO ₂	313.10	11	-0.3817	12.59	0.0839	18.86	-0.0716	14.65
2,3',4',5-TCB	CO ₂	323.10	8	-0.3673	30.40	0.0820	11.19	-0.0732	20.38
TPM	CO ₂	303.15	9	-0.3902	33.40	0.0891	22.90	-0.0699	34.04
TPM	CO ₂	313.15	6	-0.3727	27.80	0.0890	11.02	-0.0687	21.65
TPM	CO ₂	323.15	7	-0.3568	40.70	0.0901	10.59	-0.0682	26.25
TPM	C ₂ H ₆	303.15	8	-0.4665	22.79	0.0474	14.72	-0.1280	23.51
TPM	C ₂ H ₆	313.15	6	-0.4406	22.03	0.0497	12.87	-0.1222	19.24
TPM	C ₂ H ₆	323.15	6	-0.4272	17.69	0.0426	5.23	-0.1278	7.91
Overall Average Error (%)				31.06		20.46		24.28	

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Yu et al. (1995)

Yu et al. (1995)

Yu et al. (1995)

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Johnston et al. (1982)

Table 4.5: EOS Modeling of Solubility: Optimum $k_{1,2}$ Using the
Cubic Root Mixing Rule (CMR)

Equation of State			S-R-K		P-R		R-K		References
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	
Anthracene	CHF ₃	328.15	3	0.1444	74.56	0.2008	34.22	0.1738	Schmitt and Reid (1986)
Anthracene	CHF ₃	343.15	3	0.1552	68.81	0.1866	23.85	0.1675	Schmitt and Reid (1986)
Anthracene	CO ₂	303.15	4	-0.0949	70.37	0.0430	64.04	-0.0014	Johnston et al. (1982)
Anthracene	CO ₂	308.00	5	-0.0895	51.37	0.0403	41.02	-0.0042	Kosal and Holder (1987)
Anthracene	CO ₂	318.00	6	-0.0732	44.43	0.0372	21.21	-0.0036	Kosal and Holder (1987)
Anthracene	CO ₂	323.15	10	-0.0737	83.92	0.0216	50.77	-0.0159	Johnston and Eckert (1981)
Anthracene	CO ₂	343.15	9	-0.0368	97.78	0.0086	42.86	-0.0171	Johnston and Eckert (1981)
Anthracene	C ₂ H ₆	303.15	6	0.0218	61.10	0.1586	52.03	0.1066	Johnston et al. (1982)
Anthracene	C ₂ H ₆	308.15	4	0.0349	65.27	0.1626	54.81	0.1117	Johnston et al. (1982)
Anthracene	C ₂ H ₆	323.15	10	0.0477	56.74	0.1556	41.67	0.1036	Johnston et al. (1982)
Anthracene	C ₂ H ₆	343.15	7	0.0716	53.54	0.1504	33.77	0.1003	Johnston et al. (1982)
Anthracene	C ₂ H ₄	323.15	7	-0.0504	64.09	0.0393	37.84	-0.0192	Johnston and Eckert (1981)
Anthracene	C ₂ H ₄	343.15	9	-0.0022	56.31	0.0504	26.13	-0.0024	Johnston and Eckert (1981)
Anthracene	C ₂ H ₄	358.15	13	0.0127	46.06	0.0434	20.41	-0.0061	Johnston and Eckert (1981)
Benzoic Acid	CHF ₃	318.15	6	0.1570	4.13	0.5282	15.78	0.1797	Schmitt and Reid (1986)
Benzoic Acid	CHF ₃	328.15	6	0.4709	8.45	0.5233	11.84	0.4812	Schmitt and Reid (1986)
Benzoic Acid	CHF ₃	318.25	5	0.3749	8.43	0.4494	25.12	0.4001	Schmitt and Reid (1986)
Benzoic Acid	CHF ₃	328.25	5	0.3900	4.24	0.4461	21.61	0.4037	Schmitt and Reid (1986)

continued on next page

continued from previous page

Equation of State				S-R-K		P-R		R-K		References
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)	
2,3-DMN	CO ₂	318.00	5	0.1871	2.90	0.2869	20.55	0.2416	8.26	Kurnik et al. (1981)
2,3-DMN	CO ₂	328.00	5	0.2077	10.91	0.2933	19.03	0.2523	6.55	Kurnik et al. (1981)
2,3-DMN	C ₂ H ₄	308.00	6	0.2500	24.16	0.3779	42.65	0.3118	36.52	Kurnik et al. (1981)
2,3-DMN	C ₂ H ₄	318.00	6	0.3078	41.25	0.4470	61.08	0.3800	55.29	Kurnik et al. (1981)
2,3-DMN	C ₂ H ₄	328.00	6	0.4613	71.43	0.6509	78.95	0.5730	77.32	Kurnik et al. (1981)
2,6-DMN	CO ₂	308.00	5	0.1856	7.38	0.3003	15.96	0.2509	10.74	Kurnik et al. (1981)
2,6-DMN	CO ₂	318.00	5	0.1856	12.49	0.2831	9.16	0.2383	6.89	Kurnik et al. (1981)
2,6-DMN	CO ₂	328.00	5	0.2330	3.54	0.3272	60.26	0.2830	46.53	Kurnik et al. (1981)
2,6-DMN	C ₂ H ₄	308.00	6	0.2306	14.03	0.3457	32.17	0.2853	26.02	Kurnik et al. (1981)
2,6-DMN	C ₂ H ₄	318.00	6	0.2733	25.06	0.3920	45.85	0.3291	39.44	Kurnik et al. (1981)
2,6-DMN	C ₂ H ₄	328.00	6	0.3383	44.31	0.4719	64.44	0.4047	58.76	Kurnik et al. (1981)
Fluorene	CO ₂	303.15	7	0.0739	29.79	0.2072	22.34	0.1533	32.86	Johnston et al. (1982)
Fluorene	CO ₂	308.15	6	0.0825	26.63	0.2091	15.80	0.1559	27.91	Johnston et al. (1982)
Fluorene	CO ₂	323.15	9	0.1121	36.34	0.2173	13.00	0.1693	27.28	Johnston et al. (1982)
Fluorene	CO ₂	343.15	8	0.1211	41.71	0.2012	8.60	0.1583	19.89	Johnston et al. (1982)
Fluorene	C ₂ H ₄	298.15	6	0.0978	10.57	0.2326	19.93	0.1663	12.55	Johnston et al. (1982)
Fluorene	C ₂ H ₄	318.15	7	0.1509	5.48	0.2765	37.85	0.2073	26.16	Johnston et al. (1982)
Fluorene	C ₂ H ₄	343.15	11	0.2761	43.91	0.4208	71.70	0.3456	63.16	Johnston et al. (1982)
2,2',3,3',4,4'-HCB	CO ₂	313.10	5	-0.0976	43.94	0.0753	33.77	-0.0142	42.56	Yu et al. (1995)
2,2',3,3',4,4'-HCB	CO ₂	323.10	5	-0.0806	49.28	0.0712	30.36	-0.0125	42.59	Yu et al. (1995)

continued on next page

continued from previous page

Equation of State				S-R-K		P-R		R-K		References
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)	
HMB	CO ₂	303.15	7	0.2764	59.86	0.3699	58.79	0.3323	62.33	Johnston et al. (1982)
HMB	CO ₂	323.15	10	0.3115	100.53	0.3784	86.59	0.3460	96.22	Johnston et al. (1982)
HMB	CO ₂	343.15	10	0.3216	114.66	0.3549	106.16	0.3295	118.93	Johnston et al. (1982)
HMB	C ₂ H ₄	298.15	9	0.2857	67.45	0.3852	37.41	0.3317	54.53	Johnston et al. (1982)
HMB	C ₂ H ₄	318.15	9	0.3182	66.39	0.4123	15.59	0.3528	39.83	Johnston et al. (1982)
HMB	C ₂ H ₄	343.15	6	0.3885	31.25	0.4935	29.61	0.4282	15.86	Johnston et al. (1982)
Naphthalene	CClF ₃	308.15	7	0.5096	9.56	0.5750	18.34	0.5440	12.76	Schmitt and Reid (1986)
Naphthalene	CClF ₃	318.15	7	0.5222	5.19	0.5771	20.52	0.5486	11.85	Schmitt and Reid (1986)
Naphthalene	CClF ₃	328.15	7	0.5328	3.96	0.5795	21.17	0.5531	8.35	Schmitt and Reid (1986)
Naphthalene	CHF ₃	308.15	6	0.4506	10.54	0.5188	27.46	0.4886	17.71	Schmitt and Reid (1986)
Naphthalene	CHF ₃	318.15	6	0.4640	3.60	0.5212	31.65	0.4947	14.88	Schmitt and Reid (1986)
Naphthalene	CHF ₃	328.15	6	0.4789	5.02	0.5264	15.23	0.5036	6.22	Schmitt and Reid (1986)
Naphthalene	CO ₂	308.15	9	0.3465	3.67	0.4357	3.86	0.4039	2.63	McHugh and Paulaitis (1980)
Naphthalene	CO ₂	328.15	16	0.3833	9.34	0.4687	12.29	0.4392	6.15	McHugh and Paulaitis (1980)
Naphthalene	C ₂ H ₆	293.15	7	0.4155	18.74	0.5182	14.25	0.4840	13.87	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	298.15	7	0.4299	9.99	0.5301	13.56	0.4947	10.92	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	308.15	6	0.4504	12.62	0.5471	22.18	0.5120	18.77	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	318.15	8	0.4783	10.38	0.5753	23.26	0.5397	19.48	Johnston et al. (1982)
Naphthalene	C ₂ H ₆	328.15	13	0.5329	12.89	0.6813	33.12	0.6399	28.74	Johnston et al. (1982)
Naphthalene	C ₂ H ₄	298.15	10	0.3623	15.10	0.4500	29.58	0.4131	26.17	Johnston and Eckert (1981)

continued on next page

continued from previous page

Equation of State				S-R-K		P-R		R-K		References
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)	
Naphthalene	C ₂ H ₄	318.15	5	0.4180	13.36	0.4985	30.55	0.4623	27.35	Johnston and Eckert (1981)
Naphthalene	C ₂ H ₄	323.15	4	0.4451	20.27	0.5391	40.12	0.5018	36.87	Johnston and Eckert (1981)
Phenanthrene	CClF ₃	318.15	4	0.2865	5.83	0.3757	11.85	0.3294	5.79	Schmitt and Reid (1986)
Phenanthrene	CClF ₃	328.15	4	0.3018	5.99	0.3778	15.41	0.3356	5.15	Schmitt and Reid (1986)
Phenanthrene	CHF ₃	318.15	4	0.2032	14.47	0.2972	12.03	0.2506	7.45	Schmitt and Reid (1986)
Phenanthrene	CHF ₃	328.15	4	0.2193	17.56	0.2958	13.43	0.2547	5.57	Schmitt and Reid (1986)
Phenanthrene	CO ₂	303.15	8	0.0401	17.29	0.1856	14.91	0.1262	17.61	Johnston et al. (1982)
Phenanthrene	CO ₂	308.15	7	0.0510	10.30	0.1866	11.22	0.1305	10.35	Doobs et al. (1986)
Phenanthrene	CO ₂	318.00	5	0.0678	5.78	0.1852	17.97	0.1348	9.15	Kurnik et al. (1981)
Phenanthrene	CO ₂	323.15	6	0.0745	19.24	0.1917	12.85	0.1385	11.11	Johnston et al. (1982)
Phenanthrene	CO ₂	328.00	5	0.0765	13.82	0.1723	13.05	0.1285	3.47	Kurnik et al. (1981)
Phenanthrene	CO ₂	338.00	5	0.0980	20.50	0.1737	16.03	0.1367	2.47	Kurnik et al. (1981)
Phenanthrene	CO ₂	343.15	7	0.1220	33.59	0.2076	23.66	0.1641	5.81	Johnston et al. (1982)
Phenanthrene	C ₂ H ₆	303.00	6	0.1654	6.21	0.3006	10.28	0.2437	6.25	Johnston et al. (1982)
Phenanthrene	C ₂ H ₆	313.00	4	0.1720	4.43	0.3059	14.44	0.2447	8.60	Johnston et al. (1982)
Phenanthrene	C ₂ H ₆	333.00	7	0.2189	6.71	0.3333	32.39	0.2712	20.68	Johnston et al. (1982)
Phenanthrene	C ₂ H ₄	298.15	11	0.0593	6.95	0.1907	20.56	0.1273	15.17	Johnston and Eckert (1981)
Phenanthrene	C ₂ H ₄	318.00	5	0.1012	4.29	0.2068	22.89	0.1442	16.98	Kurnik et al. (1981)
Phenanthrene	C ₂ H ₄	318.15	11	0.0890	4.49	0.1877	16.79	0.1270	11.20	Johnston and Eckert (1981)
Phenanthrene	C ₂ H ₄	328.00	5	0.1038	6.00	0.1983	19.54	0.1364	12.64	Kurnik et al. (1981)

continued on next page

<i>continued from previous page</i>									
Equation of State				S-R-K		P-R		R-K	
Solute	Solvent	Temp.	N	k _{ij}	AE(%)	k _{ij}	AE(%)	k _{ij}	AE(%)
Phenanthrene	C ₂ H ₄	335.00	5	0.1614	15.49	0.2610	39.90	0.1965	33.29
Phenanthrene	C ₂ H ₄	343.15	8	0.1415	6.27	0.2102	27.76	0.1527	19.24
Pyrene	CO ₂	308.15	7	-0.1106	63.40	0.0468	48.25	-0.0175	61.36
Pyrene	CO ₂	323.15	7	-0.0959	63.20	0.0353	33.87	-0.0252	51.35
Pyrene	CO ₂	343.15	8	-0.0714	83.12	0.0216	22.44	-0.0308	56.09
Pyrene	C ₂ H ₄	318.15	8	-0.0942	47.44	0.0368	16.65	-0.0440	31.81
Pyrene	C ₂ H ₄	348.15	7	-0.0565	50.91	0.0332	5.91	-0.0455	26.88
2,3',4',5-TCB	CO ₂	308.10	6	-0.1092	11.49	0.0576	11.83	-0.0203	11.65
2,3',4',5-TCB	CO ₂	313.10	11	-0.1160	15.49	0.0443	25.33	-0.0331	19.66
2,3',4',5-TCB	CO ₂	323.10	8	-0.1012	27.00	0.0415	14.72	-0.0327	19.23
TPM	CO ₂	303.15	9	-0.1880	32.68	0.0018	20.58	-0.0848	35.09
TPM	CO ₂	313.15	6	-0.1670	25.28	0.0040	12.72	-0.0779	22.04
TPM	CO ₂	323.15	7	-0.1491	40.70	0.0072	8.31	-0.0731	22.15
TPM	C ₂ H ₆	303.15	8	-0.0428	9.23	0.1449	10.55	0.0557	6.50
TPM	C ₂ H ₆	313.15	6	-0.0314	17.12	0.1329	10.21	0.0494	11.99
TPM	C ₂ H ₆	323.15	6	0.0003	6.09	0.1541	23.40	0.0708	16.49
Overall Average Error (%)				27.44		28.01		27.63	

References

Kurnik et al. (1981)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Yu et al. (1995)
 Yu et al. (1995)
 Yu et al. (1995)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)
 Johnston et al. (1982)

Table 4.6: EOS Modeling of Solubility: Optimum k_{12} & l_{12}
Using the Quadratic Mixing Rule (QMR)

Equation of State				S-R-K			P-R			R-K		
Solute	Solvent	Temp.	N	k_{ij}	l_{ij}	AE(%)	k_{ij}	l_{ij}	AE(%)	k_{ij}	l_{ij}	AE(%)
Anthracene	CHF ₃	328.15	3	0.1236	0.4599	4.46	0.2569	0.2704	1.87	0.2428	0.3607	2.27
Anthracene	CHF ₃	343.15	3	0.1597	0.4924	14.47	0.2298	0.2328	8.01	0.2454	0.3734	10.43
Anthracene	CO ₂	303.15	4	-0.0141	0.3570	6.53	0.2751	0.3367	6.12	0.2226	0.3594	6.45
Anthracene	CO ₂	308.00	5	0.0040	0.3788	14.15	0.2657	0.3280	13.39	0.2208	0.3634	13.34
Anthracene	CO ₂	318.00	6	0.0205	0.3761	10.66	0.2182	0.2349	5.05	0.1949	0.3115	7.16
Anthracene	CO ₂	323.15	10	0.0658	0.4245	9.31	0.2518	0.2996	4.95	0.2265	0.3663	5.81
Anthracene	CO ₂	343.15	9	0.1477	0.4983	11.35	0.2371	0.2816	6.69	0.2429	0.3950	6.86
Anthracene	C ₂ H ₆	303.15	6	-0.0901	0.3318	9.12	0.2208	0.2994	8.98	0.1656	0.3355	9.08
Anthracene	C ₂ H ₆	308.15	4	-0.0546	0.3629	8.75	0.2341	0.3185	8.40	0.1967	0.3799	9.81
Anthracene	C ₂ H ₆	323.15	10	-0.0463	0.3611	5.83	0.2112	0.2903	6.89	0.1680	0.3488	6.70
Anthracene	C ₂ H ₆	343.15	7	0.0027	0.3984	6.97	0.1981	0.2765	7.95	0.1553	0.3382	7.71
Anthracene	C ₂ H ₄	323.15	7	-0.0132	0.4129	3.43	0.1926	0.2899	3.24	0.1482	0.3480	3.08
Anthracene	C ₂ H ₄	343.15	9	0.0819	0.4944	6.77	0.1996	0.2983	9.51	0.1683	0.3752	9.02
Anthracene	C ₂ H ₄	358.15	13	0.0837	0.4778	5.44	0.1623	0.2388	6.71	0.1387	0.3332	6.48
Benzoic Acid	CClF ₃	318.15	6	-0.3323	0.0111	3.91	0.0298	-0.1020	5.88	-0.1153	-0.0149	4.77
Benzoic Acid	CClF ₃	328.15	6	-0.2545	0.1337	6.52	0.0254	-0.1072	3.10	-0.0936	0.0246	2.68
Benzoic Acid	CHF ₃	318.25	5	-0.3964	-0.0448	3.45	-0.0498	-0.2647	5.58	-0.1853	-0.1358	2.87
Benzoic Acid	CHF ₃	328.25	5	-0.3387	0.0109	5.45	-0.0553	-0.2705	4.58	-0.1753	-0.1217	2.36

continued on next page

<i>continued from previous page</i>									
Equation of State				S-R-K			P-R		
Solute	Solvent	Temp.	N	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)
Benzoic Acid	CO ₂	318.00	5	-0.3815	0.0264	4.95	-0.0403	-0.1998	4.03
Benzoic Acid	CO ₂	328.00	5	-0.1540	0.3765	5.25	0.0499	0.0477	5.52
Benzoic Acid	CO ₂	338.00	5	-0.0471	0.5272	5.17	0.0547	0.0859	0.92
Benzoic Acid	C ₂ H ₆	308.15	8	-0.4843	-0.0571	7.21	-0.0181	-0.1216	7.06
Benzoic Acid	C ₂ H ₆	318.15	8	-0.4385	-0.0337	7.29	-0.0351	-0.1716	3.99
Benzoic Acid	C ₂ H ₆	328.15	7	-0.3455	0.1020	4.39	0.0053	-0.0502	4.57
Benzoic Acid	C ₂ H ₆	343.15	7	-0.2990	0.1444	4.82	-0.0236	-0.0912	3.25
Benzoic Acid	C ₂ H ₄	318.00	5	-1.2975	-1.4745	54.21	-0.8143	-2.1461	59.09
Benzoic Acid	C ₂ H ₄	328.00	5	-0.9611	-0.9193	49.45	-0.5341	-1.3615	55.89
Benzoic Acid	C ₂ H ₄	338.00	5	-0.6498	-0.3999	42.23	-0.3471	-0.8378	46.25
Biphenyl	CO ₂	308.15	7	-0.0612	0.3810	19.10	0.1952	0.2939	18.97
Biphenyl	CO ₂	308.95	8	-0.3369	-0.0768	6.01	-0.0122	-0.1701	6.52
Biphenyl	CO ₂	318.55	8	-0.2834	-0.0036	1.39	0.0055	-0.1175	0.62
Biphenyl	CO ₂	322.65	8	-0.2635	0.0275	3.07	0.0056	-0.1054	1.30
Biphenyl	CO ₂	328.15	6	-0.1246	0.2201	5.31	0.0346	-0.0483	8.14
Biphenyl	C ₂ H ₆	308.35	5	-0.4846	-0.1692	7.37	-0.0906	-0.2192	7.67
Biphenyl	C ₂ H ₆	318.15	5	-0.1121	0.0663	3.68	0.0865	0.1941	9.11
4,4'-DCB	CO ₂	313.10	9	-0.3114	0.0583	9.76	0.0312	-0.0591	12.93
4,4'-DCB	CO ₂	323.10	8	-0.2890	0.0712	2.79	-0.0018	-0.1179	8.15
2,3-DMN	CO ₂	308.00	5	-0.2472	0.0444	3.03	0.0750	-0.0353	2.66

continued on next page

continued from previous page												
Equation of State				S-R-K			P-R			R-K		
Solute	Solvent	Temp.	N	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)
2,3-DMN	CO ₂	318.00	5	-0.2279	0.0404	5.24	0.0220	-0.1544	2.61	-0.0312	-0.0469	0.89
2,3-DMN	CO ₂	328.00	5	-0.1576	0.1169	4.32	0.0261	-0.1501	1.78	-0.0006	0.0039	1.26
2,3-DMN	C ₂ H ₄	308.00	6	-0.2632	0.1682	5.62	0.0143	-0.0029	3.16	-0.0821	0.0112	3.18
2,3-DMN	C ₂ H ₄	318.00	6	-0.1900	0.2637	4.86	0.0102	0.0033	1.65	-0.0169	0.1217	3.38
2,3-DMN	C ₂ H ₄	328.00	6	-0.1940	0.2520	6.05	-0.0216	-0.0447	2.42	-0.0969	0.0435	3.71
2,6-DMN	CO ₂	308.00	5	-0.2320	0.0618	4.53	0.0828	-0.0172	5.30	0.0067	0.0352	5.29
2,6-DMN	CO ₂	318.00	5	-0.1216	0.2116	7.07	0.0743	-0.0229	3.79	0.0212	0.0715	3.08
2,6-DMN	CO ₂	328.00	5	-0.1095	0.2070	3.51	0.0938	0.0222	8.94	0.0240	0.0755	5.43
2,6-DMN	C ₂ H ₄	308.00	6	-0.2521	0.1901	3.01	0.0110	0.0036	1.93	-0.0682	0.0795	1.78
2,6-DMN	C ₂ H ₄	318.00	6	-0.1827	0.2798	5.11	0.0246	0.0485	2.83	-0.0539	0.1206	3.03
2,6-DMN	C ₂ H ₄	328.00	6	-0.1526	0.3102	4.97	-0.0023	0.0050	1.52	-0.0595	0.1207	1.80
Fluorene	CO ₂	303.15	7	-0.2184	0.1199	10.63	0.1216	0.0745	8.39	0.0508	0.1201	10.24
Fluorene	CO ₂	308.15	6	-0.2022	0.1309	7.16	0.1048	0.0426	8.62	0.0480	0.1145	8.20
Fluorene	CO ₂	323.15	9	-0.1316	0.1987	4.48	0.1030	0.0127	4.08	0.0598	0.1358	2.40
Fluorene	CO ₂	343.15	8	-0.0442	0.2833	6.40	0.1036	0.0571	4.08	0.0899	0.1920	1.22
Fluorene	C ₂ H ₄	298.15	6	-0.2990	0.1490	6.65	0.0336	0.0410	9.56	-0.0427	0.1051	9.64
Fluorene	C ₂ H ₄	318.15	7	-0.2207	0.2167	3.19	0.0378	0.0591	9.45	-0.0299	0.1370	8.61
Fluorene	C ₂ H ₄	343.15	11	-0.1640	0.2580	4.83	0.0157	0.0370	3.40	-0.0470	0.1239	2.33
2,2',3,3',4,4'-HCB	CO ₂	313.10	5	-0.0613	0.2964	13.75	0.2906	0.2113	14.19	0.1677	0.2773	14.02
2,2',3,3',4,4'-HCB	CO ₂	323.10	5	-0.0202	0.3354	10.93	0.2977	0.2316	12.76	0.1810	0.3021	12.04

continued on next page

continued on next page

continued from previous page												
Equation of State				S-R-K			P-R			R-K		
Solute	Solvent	Temp.	N	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)
HMB	CO ₂	303.15	7	0.1674	0.3263	12.65	0.4060	0.3030	11.71	0.3343	0.3222	12.53
HMB	CO ₂	323.15	10	0.3256	0.5079	15.91	0.4641	0.4103	9.49	0.4170	0.4549	10.89
HMB	CO ₂	343.15	10	0.3629	0.5243	21.20	0.4518	0.3881	14.09	0.4260	0.4593	15.83
HMB	C ₂ H ₄	298.15	9	0.1291	0.3795	8.36	0.3575	0.3083	3.96	0.2868	0.3523	4.01
HMB	C ₂ H ₄	318.15	9	0.1758	0.4162	12.57	0.3544	0.3105	6.24	0.2899	0.3633	6.69
HMB	C ₂ H ₄	343.15	6	0.3497	0.6209	15.94	0.4013	0.4136	11.32	0.3424	0.4590	11.41
Naphthalene	CClF ₃	308.15	7	-0.2527	-0.0394	5.03	0.0227	-0.1252	6.69	-0.0192	-0.0641	6.62
Naphthalene	CClF ₃	318.15	7	-0.2251	-0.0226	4.39	0.0059	-0.1592	5.30	-0.0208	-0.0681	4.53
Naphthalene	CClF ₃	328.15	7	-0.1843	0.0227	5.32	0.0050	-0.1514	3.04	-0.0137	-0.0480	1.37
Naphthalene	CHF ₃	308.15	6	-0.2434	-0.0466	3.45	0.0058	-0.1743	8.62	-0.0218	-0.0892	6.58
Naphthalene	CHF ₃	318.15	6	-0.1960	0.0072	4.11	-0.0014	-0.1814	7.60	-0.0123	-0.0661	4.55
Naphthalene	CHF ₃	328.15	6	-0.1558	0.0483	6.55	-0.0106	-0.1954	5.34	-0.0043	-0.0494	3.32
Naphthalene	CO ₂	328.15	16	-0.1050	0.1699	5.66	0.0451	-0.0394	5.70	0.0043	-0.0003	4.50
Naphthalene	CO ₂	308.15	9	-0.2155	0.0091	2.37	0.0468	-0.0736	3.43	0.0106	-0.0196	3.26
Naphthalene	C ₂ H ₆	293.15	7	-0.5176	-0.3593	6.90	-0.0727	-0.2519	7.81	-0.1407	-0.2416	7.26
Naphthalene	C ₂ H ₆	298.15	7	-0.3603	-0.0884	6.64	-0.0067	-0.0936	4.56	-0.0613	-0.0577	5.44
Naphthalene	C ₂ H ₆	308.15	6	-0.3843	-0.1382	6.01	-0.0607	-0.1927	6.07	-0.1167	-0.1534	6.59
Naphthalene	C ₂ H ₆	318.15	8	-0.3380	-0.0806	3.38	-0.0739	-0.2127	5.43	-0.1194	-0.1495	4.76
Naphthalene	C ₂ H ₆	328.15	13	-0.2873	0.0068	2.55	-0.0295	-0.0989	5.60	-0.0646	-0.0282	5.62
Naphthalene	C ₂ H ₄	298.15	10	-0.3617	-0.0651	2.91	-0.1206	-0.2980	6.04	-0.1628	-0.2238	5.47
continued on next page												

continued on next page

continued from previous page												
Equation of State				S-R-K			P-R			R-K		
Solute	Solvent	Temp.	N	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)
Phenanthrene	C ₂ H ₄	338.00	5	-0.0091	0.5265	2.29	0.1037	0.1913	4.69	0.0408	0.2813	4.32
Phenanthrene	C ₂ H ₄	343.15	8	-0.0831	0.3816	5.07	0.0280	-0.0023	4.63	-0.0287	0.1194	4.56
Pyrene	CO ₂	308.15	7	-0.0972	0.2774	11.91	0.2434	0.2228	12.71	0.1582	0.2621	13.14
Pyrene	CO ₂	323.15	7	-0.0506	0.3161	4.85	0.2252	0.1960	5.67	0.1610	0.2719	3.70
Pyrene	CO ₂	343.15	8	0.0157	0.3708	6.44	0.1992	0.1569	3.80	0.1614	0.2765	1.04
Pyrene	C ₂ H ₄	318.15	8	-0.1822	0.2840	3.76	0.1276	0.1437	6.60	0.0434	0.2244	6.27
Pyrene	C ₂ H ₄	348.15	7	-0.0796	0.3709	3.62	0.1251	0.1591	4.60	0.0371	0.2308	3.61
2,3',4',5-TCB	CO ₂	308.10	6	-0.3182	0.0750	7.20	0.0963	0.0019	7.82	-0.0245	0.0604	8.09
2,3',4',5-TCB	CO ₂	313.10	11	-0.3548	0.0370	11.76	0.0428	-0.0840	15.66	-0.0744	-0.0050	14.63
2,3',4',5-TCB	CO ₂	323.10	8	-0.2401	0.1677	6.10	0.0941	0.0235	10.89	-0.0072	0.1119	8.90
TPM	CO ₂	303.15	9	-0.2758	0.1423	9.92	0.1436	0.1003	9.27	0.0198	0.1449	9.80
TPM	CO ₂	313.15	6	-0.2522	0.1552	3.60	0.1005	0.0222	8.81	-0.0052	0.1059	6.92
TPM	CO ₂	323.15	7	-0.2202	0.1701	14.14	0.1041	0.0256	9.90	0.0044	0.1152	11.12
TPM	C ₂ H ₆	303.15	8	-0.3585	0.1371	7.69	0.0952	0.0915	6.48	-0.0594	0.1136	8.54
TPM	C ₂ H ₆	313.15	6	-0.3243	0.1616	6.71	0.0929	0.0898	7.57	-0.0490	0.1315	7.63
TPM	C ₂ H ₆	323.15	6	-0.3148	0.1668	6.79	0.0412	-0.0031	5.23	-0.0940	0.0649	5.23
Overall Average Error (%)						7.63	7.62			7.28		

Table 4.7: EOS Modeling of Solubility: Optimum k_{12} & l_{12}
Using the Cubic Root Mixing Rule (CMR)

Equation of State				S-R-K			P-R			R-K		
Solute	Solvent	Temp.	N	k_{ij}	l_{ij}	AE(%)	k_{ij}	l_{ij}	AE(%)	k_{ij}	l_{ij}	AE(%)
Anthracene	CHF ₃	328.15	3	0.4031	0.4326	3.18	0.3448	0.2566	3.63	0.3714	0.3476	0.61
Anthracene	CHF ₃	343.15	3	0.4356	0.4653	13.51	0.2979	0.1999	7.37	0.3617	0.3441	9.52
Anthracene	CO ₂	303.15	4	0.1433	0.3323	7.65	0.2409	0.3020	6.60	0.2164	0.3324	7.51
Anthracene	CO ₂	308.00	5	0.1529	0.3420	11.99	0.2193	0.2797	10.82	0.2068	0.3245	11.01
Anthracene	CO ₂	318.00	6	0.1807	0.3536	8.58	0.1566	0.1857	2.88	0.1727	0.2676	4.69
Anthracene	CO ₂	323.15	10	0.2221	0.3976	7.20	0.2091	0.2706	5.71	0.2258	0.3452	5.12
Anthracene	CO ₂	343.15	9	0.3044	0.4703	9.81	0.1782	0.2450	7.37	0.2401	0.3658	6.57
Anthracene	C ₂ H ₆	303.15	6	0.2344	0.3167	9.19	0.3275	0.2808	8.88	0.3057	0.3234	9.06
Anthracene	C ₂ H ₆	308.15	4	0.2662	0.3486	9.89	0.3433	0.3030	9.64	0.3232	0.3465	10.22
Anthracene	C ₂ H ₆	323.15	10	0.2829	0.3587	6.02	0.3184	0.2763	7.51	0.3054	0.3311	7.41
Anthracene	C ₂ H ₆	343.15	7	0.3135	0.3763	6.76	0.3057	0.2662	8.34	0.2890	0.3124	7.72
Anthracene	C ₂ H ₄	323.15	7	0.2347	0.3940	2.97	0.2391	0.3031	5.67	0.2070	0.3311	3.24
Anthracene	C ₂ H ₄	343.15	9	0.3151	0.4623	6.71	0.2099	0.2582	9.44	0.2187	0.3402	8.98
Anthracene	C ₂ H ₄	358.15	13	0.2987	0.4179	5.22	0.1738	0.2110	6.67	0.1865	0.2964	6.46
Benzoic Acid	CClF ₃	318.15	6	0.4599	0.0076	4.01	0.4806	-0.1458	7.99	0.1611	-0.0443	6.48
Benzoic Acid	CClF ₃	328.15	6	0.5005	0.0850	4.63	0.4731	-0.1648	4.44	0.4671	-0.0429	2.79
Benzoic Acid	CHF ₃	318.25	5	0.3399	-0.0877	1.65	0.3420	-0.3107	7.80	0.3366	-0.1704	5.12
Benzoic Acid	CHF ₃	328.25	5	0.3784	-0.0292	3.36	0.3359	-0.3185	6.14	0.3472	-0.1509	3.37

continued on next page

continued from previous page									
Equation of State				S-R-K			P-R		
Solute	Solvent	Temp.	N	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)
Benzoic Acid	CO ₂	318.00	5	0.2026	-0.0612	3.57	0.2440	-0.2277	6.47
Benzoic Acid	CO ₂	328.00	5	0.3029	0.1282	4.47	0.2400	-0.2355	5.68
Benzoic Acid	CO ₂	338.00	5	0.3663	0.2381	6.28	0.2208	-0.2890	0.41
Benzoic Acid	C ₂ H ₆	308.15	8	0.2963	-0.1060	5.82	0.3827	-0.1867	5.50
Benzoic Acid	C ₂ H ₆	318.15	8	0.3258	-0.0841	4.22	0.3753	-0.2265	2.37
Benzoic Acid	C ₂ H ₆	328.15	7	0.3789	0.0286	3.95	0.3988	-0.1509	6.27
Benzoic Acid	C ₂ H ₆	343.15	7	0.4050	0.0435	3.53	0.3683	-0.2504	6.65
Benzoic Acid	C ₂ H ₄	318.00	5	-0.2061	-1.2353	55.69	-0.1957	-1.6281	64.45
Benzoic Acid	C ₂ H ₄	328.00	5	-0.0523	-0.8841	51.49	-0.1402	-1.4728	62.69
Benzoic Acid	C ₂ H ₄	338.00	5	0.0968	-0.5327	44.75	-0.0262	-1.1189	52.71
Biphenyl	CO ₂	308.15	7	0.2164	-0.0494	4.59	0.2503	-0.2220	3.81
Biphenyl	CO ₂	308.95	8	0.2338	-0.0182	5.03	0.3182	-0.0746	5.29
Biphenyl	CO ₂	318.55	8	0.2887	0.0402	2.86	0.3345	-0.0719	5.28
Biphenyl	CO ₂	322.65	8	0.3082	0.0534	4.03	0.3577	-0.0502	7.77
Biphenyl	CO ₂	328.15	6	0.2855	-0.0081	1.26	0.2783	-0.2113	5.07
Biphenyl	C ₂ H ₆	308.35	5	0.2664	-0.1923	7.33	0.3382	-0.3241	7.90
Biphenyl	C ₂ H ₆	318.15	5	0.3063	-0.1748	9.24	0.3372	-0.3875	10.91
4,4'-DCB	CO ₂	313.10	9	-0.0020	0.0053	12.76	0.0688	-0.1019	17.13
4,4'-DCB	CO ₂	323.10	8	0.0326	0.0388	3.86	0.0203	-0.1785	12.12
2,3-DMN	CO ₂	308.00	5	0.1567	-0.0525	2.74	0.2253	-0.1596	4.71

continued on next page

Equation of State

continued on next page

continued on next page

<i>continued from previous page</i>									
Equation of State				S-R-K			P-R		
Solute	Solvent	Temp.	N	k _{ij}	l _{ij}	AE(%)	k _{ij}	l _{ij}	AE(%)
Phenanthrene	C ₂ H ₆	313.00	4	0.1634	-0.0155	3.44	0.2214	-0.1756	3.03
Phenanthrene	C ₂ H ₆	333.00	7	0.2031	-0.0283	5.11	0.2190	-0.2277	8.21
Pyrene	CO ₂	308.15	7	0.0694	0.2495	15.26	0.1667	0.1850	16.66
Pyrene	CO ₂	323.15	7	0.1157	0.2856	3.28	0.1424	0.1615	8.52
Pyrene	CO ₂	343.15	8	0.1751	0.3275	4.22	0.0955	0.1074	6.06
Pyrene	C ₂ H ₄	318.15	8	0.0788	0.2450	4.01	0.0864	0.0802	7.34
Pyrene	C ₂ H ₄	348.15	7	0.1267	0.2553	1.76	0.0366	0.0054	5.81
2,3',4',5-TCB	CO ₂	308.10	6	-0.0806	0.0415	9.69	0.0359	-0.0360	10.91
2,3',4',5-TCB	CO ₂	313.10	11	-0.1062	0.0140	15.66	-0.0277	-0.1191	20.39
2,3',4',5-TCB	CO ₂	323.10	8	-0.0146	0.1197	8.35	0.0283	-0.0207	14.65
TPM	CO ₂	303.15	9	-0.1020	0.1064	12.14	0.0430	0.0581	10.60
TPM	CO ₂	313.15	6	-0.0854	0.1044	7.28	0.0040	0.0000	12.72
TPM	CO ₂	323.15	7	-0.0471	0.1255	9.77	0.0035	-0.0052	6.43
TPM	C ₂ H ₆	303.15	8	0.0044	0.0854	10.32	0.1356	0.0124	9.29
TPM	C ₂ H ₆	313.15	6	0.0186	0.0786	7.65	0.1121	-0.0374	8.59
TPM	C ₂ H ₆	323.15	6	0.0036	0.0312	4.03	0.0358	-0.1867	5.24
Overall Average Error (%)				7.67			9.60		
							9.31		

Table 4.8: EOS Modeling of Solubility: Optimum Solute to Solvent Parameter Ratios α & β in the Simplified Method

Equation of State			S-R-K				P-R			R-K		
Solute	Solvent	Temp.	N	α	β	AE(%)	α	β	AE(%)	α	β	AE(%)
Acridine ¹	CClF ₃	318.15	5	4.1456	3.2721	12.08	4.4794	4.1161	19.60	4.4078	3.6376	17.17
Acridine	CClF ₃	328.15	5	3.9892	2.9826	6.26	4.6487	4.3592	22.08	4.4599	3.6740	16.72
Acridine	CHF ₃	318.15	6	4.3478	3.1648	21.06	5.2697	5.2669	27.05	4.9254	4.1761	24.66
Acridine	CHF ₃	328.15	6	4.4354	3.4765	14.45	5.7511	6.3306	28.78	5.2475	4.9156	25.17
Acridine	CO ₂	308.25	6	5.1529	4.7258	9.30	5.2377	5.3108	8.06	5.2722	4.9462	8.18
Acridine	CO ₂	318.25	8	4.9643	4.3303	11.50	5.8323	6.6496	10.72	5.5316	5.4974	5.65
Acridine	CO ₂	328.15	7	4.4849	3.2615	16.04	6.0596	7.0874	10.27	5.5277	5.4420	7.86
Acridine	CO ₂	343.15	7	4.4168	3.2505	5.17	6.2522	7.3174	11.83	5.6222	5.6100	10.18
Acridine	C ₂ H ₆	308.25	6	4.9565	4.5479	13.65	4.9511	4.9411	12.65	5.0437	4.7094	12.31
Acridine	C ₂ H ₆	318.15	10	4.4499	3.4464	37.41	4.8715	4.6411	35.71	4.8592	4.1807	37.00
Acridine	C ₂ H ₆	328.15	9	4.6119	3.7163	24.05	5.4083	5.6921	28.66	5.3346	5.0623	30.73
Acridine	C ₂ H ₆	343.15	7	4.3587	3.2100	21.14	5.4378	5.5360	20.26	5.3100	4.8247	22.04
Anthracene	CHF ₃	328.15	3	3.6307	2.6490	9.52	4.5782	4.6165	4.91	4.1983	3.5790	1.55
Anthracene	CHF ₃	343.15	3	2.8579	1.0087	26.28	4.1024	3.2775	18.75	3.5849	2.1242	21.95
Anthracene	CO ₂	303.15	4	5.3206	5.8365	6.48	5.2230	6.0804	6.72	5.2998	5.8012	6.53
Anthracene	CO ₂	308.00	5	4.7084	4.2535	21.60	4.7828	4.7867	20.22	4.8283	4.4928	20.44
Anthracene	CO ₂	318.00	6	4.1782	3.0912	28.91	5.1202	5.6324	16.86	4.7738	4.3929	22.43

continued on next page

continued on next page

¹Reference for Exptl. Data for all Systems with this Solute is Schmitt & Reid, 1986

continued from previous page									
Equation of State				S-R-K			P-R		
Solute	Solvent	Temp.	N	α	β	AE(%)	α	β	AE(%)
Anthracene	CO ₂	323.15	10	3.9302	2.8309	43.42	5.1219	5.6561	27.39
Anthracene	CO ₂	343.15	9	4.1135	3.4556	14.30	5.6982	6.8800	13.17
Anthracene	C ₂ H ₆	303.15	6	4.6514	3.9506	7.74	4.5744	4.2343	7.63
Anthracene	C ₂ H ₆	308.15	4	4.6039	3.9299	12.23	4.5817	4.2672	11.71
Anthracene	C ₂ H ₆	323.15	10	4.4753	3.7570	7.33	4.6836	4.3777	10.00
Anthracene	C ₂ H ₆	343.15	7	4.2401	3.4261	9.48	4.7658	4.4527	11.79
Anthracene	C ₂ H ₄	323.15	7	4.6688	3.6157	3.62	5.3810	4.9225	5.73
Anthracene	C ₂ H ₄	343.15	9	4.2555	3.0108	7.87	5.6541	5.5651	8.79
Anthracene	C ₂ H ₄	358.15	13	4.4322	3.5702	13.41	5.9416	6.1034	17.13
Benzoic Acid	CClF ₃	318.15	6	2.9990	1.8590	9.86	3.3107	2.5936	6.46
Benzoic Acid	CClF ₃	328.15	6	2.4951	0.8316	25.28	3.2070	2.3142	10.46
Benzoic Acid	CHF ₃	318.25	5	3.3877	2.3041	16.43	4.2154	4.2019	11.78
Benzoic Acid	CHF ₃	328.25	5	3.0599	1.6979	19.96	3.9847	3.5711	14.39
Benzoic Acid	CO ₂	318.00	5	4.2588	3.6032	4.86	4.8132	5.1304	3.55
Benzoic Acid	CO ₂	328.00	5	3.1584	0.7644	14.55	4.4160	3.8481	5.23
Benzoic Acid	C ₂ H ₆	308.15	8	3.5999	2.4294	14.73	3.5695	2.6597	13.91
Benzoic Acid	C ₂ H ₆	318.15	8	3.1960	1.7063	29.73	3.6544	2.8866	9.19
Benzoic Acid	C ₂ H ₆	328.15	7	3.2315	1.8053	30.75	3.7262	2.9534	24.78
Benzoic Acid	C ₂ H ₆	343.15	7	2.7446	0.7996	24.53	3.3915	2.0429	23.70
Benzoic Acid	C ₂ H ₄	318.00	5	4.2370	3.3099	39.16	4.8095	4.4605	40.20

continued on next page

continued on next page

[illegible]

continued on next page

<i>continued from previous page</i>									
Equation of State				S-R-K			P-R		
Solute	Solvent	Temp.	N	α	β	AE(%)	α	β	R-K $\alpha \quad \beta \quad AE(\%)$
Naphthalene	C ₂ H ₆	293.15	7	4.2856	4.8758	37.11	3.7263	3.9059	3.8766 3.9564 11.47
Naphthalene	C ₂ H ₆	298.15	7	3.8201	3.6228	9.97	3.6201	3.5549	3.7110 3.4459 7.97
Naphthalene	C ₂ H ₆	308.15	6	4.0880	4.3471	17.98	4.0920	4.6696	4.1419 4.4502 18.12
Naphthalene	C ₂ H ₆	318.15	8	3.5666	2.9939	17.19	3.9765	4.1870	-1.0212 3.9782 13.39
Naphthalene	C ₂ H ₆	328.15	13	3.3702	2.2626	25.64	3.9195	3.5509	3.9457 3.2909 16.92
Naphthalene	C ₂ H ₄	298.15	10	4.2330	4.0415	7.73	4.7404	5.5418	4.7481 5.1787 9.48
Naphthalene	C ₂ H ₄	318.15	5	3.2874	1.2775	5.08	4.6517	4.7459	4.5180 4.0293 8.91
1,4-Naphthaquinone ²	CClF ₃	318.15	6	3.4815	2.8676	8.10	3.8359	3.7412	3.7473 3.2777 13.51
1,4-Naphthaquinone	CClF ₃	328.15	6	3.0851	2.1004	11.30	3.7251	3.4131	3.5482 2.8219 9.95
1,4-Naphthaquinone	CHF ₃	318.15	6	4.2671	3.4005	12.91	5.0451	5.1977	4.7690 4.2744 24.00
1,4-Naphthaquinone	ClIF ₃	328.15	6	4.1870	3.3386	15.31	5.4306	6.0302	4.9674 4.7295 23.21
1,4-Naphthaquinone	CO ₂	318.15	6	4.9287	4.8210	4.24	5.6116	6.6884	5.3837 5.7370 14.87
1,4-Naphthaquinone	CO ₂	328.15	6	4.2906	3.3896	9.55	5.6313	6.6479	5.1828 5.2376 10.61
1,4-Naphthaquinone	CO ₂	343.15	6	4.0004	2.8522	16.25	5.6192	6.3613	5.1053 4.9876 8.89
1,4-Naphthaquinone	C ₂ H ₆	308.15	7	3.8434	3.2975	11.57	3.8303	3.5771	10.99 3.8983 10.40
1,4-Naphthaquinone	C ₂ H ₆	318.15	7	3.6622	2.9592	7.52	3.9894	3.8931	9.70 3.9857 7.43
1,4-Naphthaquinone	C ₂ H ₆	328.15	7	3.6710	3.0590	5.00	4.2508	4.4696	4.40 4.2091 13.50
1,4-Naphthaquinone	C ₂ H ₆	343.15	7	3.4344	2.7098	11.40	4.1614	4.2484	15.27 4.0997 16.65
Phenanthrene	CClF ₃	318.15	4	3.9986	3.2687	1.79	4.3054	4.0518	4.2388 3.5992 4.77

continued on next page

continued on next page

²Reference for Exptl. Data for all Systems with this Solute is Schmitt & Reid, 1986

<i>continued from previous page</i>									
Equation of State				S-R-K			P-R		
Solute	Solvent	Temp.	N	α	β	AE(%)	α	β	AE(%)
Phenanthrene	CClF ₃	328.15	4	3.8182	3.0459	5.14	4.4089	4.2848	7.35
Phenanthrene	CHF ₃	318.15	4	4.2180	3.5447	11.44	4.8852	5.0576	7.63
Phenanthrene	CHF ₃	328.15	4	3.8887	2.9762	19.47	5.0290	5.3248	8.21
Phenanthrene	CO ₂	303.15	8	5.7074	6.6677	12.34	5.5807	6.8562	8.91
Phenanthrene	CO ₂	308.15	7	5.4989	6.2465	5.80	5.6003	6.8826	7.21
Phenanthrene	CO ₂	318.00	5	5.3623	6.0104	4.51	5.9560	7.7055	8.98
Phenanthrene	CO ₂	323.15	6	4.5921	4.2830	24.74	5.7633	7.1316	11.61
Phenanthrene	CO ₂	328.00	5	4.7103	4.4132	8.69	6.0788	7.9105	6.29
Phenanthrene	CO ₂	338.00	5	4.3893	3.7640	6.06	6.1499	8.0389	10.81
Phenanthrene	CO ₂	343.15	7	3.5101	2.1311	40.44	4.6648	4.4074	43.71
Phenanthrene	C ₂ H ₆	303.00	6	4.7986	4.3228	8.38	4.6543	4.4538	7.12
Phenanthrene	C ₂ H ₆	313.00	4	4.7683	4.1128	3.26	4.8199	4.5522	3.26
Phenanthrene	C ₂ H ₆	333.00	7	4.4367	3.7305	7.60	5.0897	5.2443	18.14
Phenanthrene	C ₂ H ₄	298.15	11	5.3978	4.7123	25.71	6.1450	6.7802	28.45
Phenanthrene	C ₂ H ₄	318.00	5	4.9207	3.8317	0.83	5.8843	5.9218	6.89
Phenanthrene	C ₂ H ₄	318.15	11	4.8561	3.7094	17.11	6.1818	6.7304	18.01
Phenanthrene	C ₂ H ₄	328.00	5	4.3338	2.4276	5.71	5.5831	4.9581	13.25
Phenanthrene	C ₂ H ₄	338.00	5	4.1336	2.0960	9.19	5.5863	4.8947	16.19
Phenanthrene	C ₂ H ₄	343.15	8	4.2608	2.6006	20.30	5.8988	5.7884	13.07
Pyrene	CO ₂	308.15	7	5.8560	6.4384	10.18	5.9995	7.2112	18.81
<i>continued on next page</i>									
				R-K					
				α	β	AE(%)			
				4.2320	3.6408	7.35			
				4.6313	4.2078	7.63			
				4.6021	4.1698	8.21			
				5.6773	6.6086	8.91			
				5.5995	6.4332	7.21			
				5.7471	6.7781	8.98			
				5.3537	5.8042	11.61			
				5.5547	6.2260	6.29			
				5.5297	6.2176	10.81			
				4.3949	3.6812	43.71			
				4.7552	4.2574	7.12			
				4.8712	4.2347	3.26			
				5.0382	4.7132	18.14			
				6.0929	6.1072	28.45			
				5.7763	5.1932	6.89			
				6.0435	5.9042	18.01			
				5.4367	4.1986	13.25			
				5.3971	4.0797	16.19			
				5.7341	5.0438	13.07			
				5.9898	6.6941	18.81			

continued from previous page												
Equation of State				S-R-K			P-R			R-K		
Solute	Solvent	Temp.	N	α	β	AE(%)	α	β	AE(%)	α	β	AE(%)
Pyrene	CO ₂	323.15	7	4.9869	4.5860	26.18	6.1260	7.3197	7.76	5.7381	6.0241	7.74
Pyrene	CO ₂	343.15	8	4.2848	3.2295	37.06	5.7159	6.0613	38.42	5.3327	5.0334	32.25
Pyrene	C ₂ H ₄	318.15	8	5.4092	4.2072	8.91	6.2928	5.9607	14.60	6.2206	5.2940	12.57
Pyrene	C ₂ H ₄	348.15	7	4.9852	3.6425	4.85	6.3859	5.8906	12.46	6.2618	5.2295	10.30
2,3',4',5-TCB	CO ₂	308.10	6	6.5100	7.9121	8.23	6.6371	8.6818	9.35	6.6113	8.0938	9.67
2,3',4',5-TCB	CO ₂	313.10	5	5.3177	5.5846	16.99	5.4708	6.2225	17.21	5.4299	5.7129	17.25
2,3',4',5-TCB	CO ₂	323.10	5	5.1861	5.3702	17.96	5.6640	6.5567	21.71	5.4989	5.8227	20.28
Triphenylmethane	CO ₂	303.15	9	6.6361	8.0369	31.84	6.5817	8.4892	29.79	6.6190	8.0127	31.60
Triphenylmethane	CO ₂	313.15	6	6.4054	7.6130	9.92	7.1685	9.9209	26.84	7.0211	9.0362	22.83
Triphenylmethane	CO ₂	323.15	7	5.6954	6.2008	24.47	6.8280	8.9866	14.00	6.4678	7.7116	11.48
Triphenylmethane	C ₂ H ₆	303.15	8	5.8706	5.6013	11.44	5.7369	5.8669	8.99	5.8271	5.5408	10.72
Triphenylmethane	C ₂ H ₆	313.15	6	5.7281	5.4740	16.95	5.8595	6.1782	20.25	5.9053	5.7602	20.29
Triphenylmethane	C ₂ H ₆	323.15	6	5.1568	4.0553	11.13	5.9790	6.3217	14.98	5.9129	5.6302	12.95
Overall Average Error (%)						14.17	14.35			13.44		

4.4 Conclusions

Several combinations of cubic equations of state and mixing rules have been studied and optimum binary interaction parameters are computed for modeling solid-supercritical fluid equilibria. A marked improvement is observed in the accuracy of the calculated solubilities when the second BIP, l_{12} , is included in the analysis. However the quadratic and cube-root mixing rules show no substantial difference when the overall percent average errors are compared. Furthermore, the three EOS tested also give similar results in terms of overall percent average errors. However, differences among the EOS are found when the errors for individual systems are compared. The following recommendation emerges from these conclusions: any of the three EOS along with any of the two mixing rules tested can be used, as long as two binary interaction parameters are included, one for the cohesive energy parameter, a , and one for the repulsive energy parameter, b . From a practical point of view, the simplest EOS and mixing rule may be preferred.

Promising results have been obtained also with a model which is considerably simpler than the rigorous EOS approach since no iteration is required to find the solute fugacity coefficient. Each of the three EOS studied gave fairly accurate predictions of solubility. The simplified model should prove to be an efficient and easy way to correlate high-molecular-weight solids solubilities, since it does not require the critical properties of the solute.

Lastly, the unified approach adopted in the present work to integrate different two-parameter cubic EOS with different mixing rules should be especially advantageous in the case of multiparameter cubic equations of state (three and four parameters) while deriving the composition dependence of the fugacity coefficient of a component in a mixture. There are many established cubic equations of state (Mika, 1989; Lawal and van der Laan, 1994) whose utility has not been extensively tested because of the lack of a cohesive technique of generalizing the composition dependence of the component fugacities for different combinations of cubic EOS and mixing rules. It is felt that such a unified approach should be of great utility in testing the capabilities of various EOS/mixing rule combinations in correlating and predicting solid-supercritical fluid equilibria.

Bibliography

- [1] ADACHI, Y., AND LU, B. C. Y. Supercritical Fluid Extraction with Carbon Dioxide and Ethylene. *Fluid Phase Equilibria* 14 (1983), 147-156.
- [2] AKGERMAN, A., ERKEY, C., AND GHOREISHI, S. M. Supercritical Extraction of Hexachlorobenzene from Soil. *Ind. Eng. Chem. Res.* 31 (1992), 333-339.
- [3] BALBUENA, P., CAMPANELLA, E., AND GRIBAUDO, L. Supercritical Phase Behavior: Description of Binary Solid Mixtures in a Supercritical Solvent with a Lattice Gas Model. *Fluid Phase Equilibria* 62 (1991), 225-238.
- [4] BAMBERGER, A., AND MAURER, G. High-Pressure Vapor-Liquid Equilibria in Binary Mixtures of Carbon Dioxide and Aromatic Hydrocarbons: Experimental Data and Correlation for CO_2 + Acetophenone, CO_2 + 1-Chloronaphthalene, CO_2 + Methyl Benzoate and CO_2 + n-Propylbenzene. *J. Supercrit. Fluids* 7 (1994), 115-127.
- [5] BAMBERGER, A., SCHMELZER, J., WALTHER, D., AND MAURER, G. High-Pressure Vapour-Liquid Equilibria in Binary Mixtures of Carbon Dioxide and Benzene Compounds: Experimental Data for Mixtures with Ethylbenzene, Isopropylbenzene, 1,2,4-Trimethylbenzene, 1,3,5-Trimethylbenzene, Ethenylbenzene and Isopropenylbenzene, and their Correlation with the Generalized Bender and Skjold-Jorgensen's Group Contribution Equation of State. *Fluid Phase Equilibria* 97 (1994), 167-189.
- [6] BARNA, L. R., RAUZY, E., BERRO, C., AND BLANCHARD, J. M. An "Excess Function-Equation of State" Model for Solubility of Hydrocarbon Solids in Supercritical Carbon Dioxide. *Fluid Phase Equilibria* 100 (1994), 191-208.

- [7] BARTLE, K. D., CLIFFORD, A. A., AND SHILSTONE, G. F. Proc. 2nd Symp. Supercritical Fluids. p. 487. 1991.
- [8] BRADY, B. O., KAO, C. C.-P., GAMBRELL, R. P., DOOLEY, K. M., AND KNOFF, F. C. Supercritical Extraction of Toxic Organics from Soil. *Ind. Eng. Chem. Res.* 26 (1987), 216-218.
- [9] BRENNECKE, J. F., AND ECKERT, C. A. Phase Equilibria for Supercritical Fluid Process Design. *AIChE J.* 35 (1989), 1409-1427.
- [10] BRUNO, T. J., AND ELY, J. F., ^{SOB} *Supercritical Fluid Technology: Reviews in Modern Theory and Applications*. CRC Press, Boca Raton, FL. 1991.
- [11] CABALLERO, A. C., AND ESTÉVEZ, L. A. Comments on "A Comparative Study of Mixing Rules in the Prediction of Solid-Vapor Equilibria". *Ind. Eng. Chem. Res.* 30 (1991), 601-605.
- [12] CABALLERO, A. C., AND ESTÉVEZ, L. A. Additions and Corrections: Comments on "A Comparative Study of Mixing Rules in the Prediction of Solid-Vapor Equilibria". *Ind. Eng. Chem. Res.* 31 (1992), 2256.
- [13] CABALLERO, A. C., HERNANDEZ, L. N., AND ESTÉVEZ, L. A. Calculation of Interaction Parameters for Binary Solid-SCF Equilibria Using Several EOS and Mixing Rules. *J. Supercrit. Fluids* 5 (1992), 283-295.
- [14] CHEN, P. C., TANG, M., AND CHEN, Y. P. A Mixing Model for the Calculation of the Solid Solubilities of Solids and Their Mixtures in Supercritical Fluids. In *Separation Technology*. Elsevier, Amsterdam, 1994, pp. 49-58.
- [15] CHEN, P. C., TANG, M., AND CHEN, Y. P. Calculations of the Solubilities of Solids in Supercritical Fluids Using the Peng-Robinson Equation of State and a Modified Mixing Model. *Ind. Eng. Chem. Res.* 34 (1995), 332-339.
- [16] CHRASTIL, J. Solubility of Solids and Liquids in Supercritical Gases. *J. Phys. Chem.* 86 (1982), 3016-3021.
- [17] CHUEH, P. L., AND PRAUSNITZ, J. M. Third Virial Coefficients of Nonpolar Gases and Their Mixtures. *AIChE J.* 13 (1967), 896-902.

- [18] COUTSIKOS, P., MAGOULAS, K., AND TASSIOS, D. Solubilities of Phenols in Supercritical Carbon Dioxide. *J. Chem. Eng. Data* 40 (1995), 953-958.
- [19] DEAK, A., AND KEMENY, S. Role of Weighting in Parameter Estimation from Solid-Fluid Supercritical Equilibrium Data. *Fluid Phase Equilibria* 100 (1994), 171-190.
- [20] DOBBS, J. M., AND JOHNSTON, K. P. Selectivities in Pure and Mixed Supercritical Fluid Solvents. *Ind. Eng. Chem. Res.* 26 (1987), 1476-1485.
- [21] DOBBS, J. M., WONG, J. M., LAHIERE, R. J., AND JOHNSTON, K. P. Modification of Supercritical Fluid Phase Behaviour Using Polar Cosolvents. *Ind. Eng. Chem. Res.* 26 (1987), 56-65.
- [22] DOOLEY, K. M., GHONASGI, D., AND KNOPF, F. C. Supercritical CO₂-Cosolvent Extraction of Contaminated Soils. *Environ. Prog.* 9 (1990), 197-203.
- [23] DOOLEY, K. M., KAO, C. C.-P., GAMBRELL, R. P., AND KNOPF, F. C. The Use of Entrainers in the Supercritical Extraction of Soils Contaminated with Hazardous Organics. *Ind. Eng. Chem. Res.* 26 (1987), 2058-2062.
- [24] ESTÉVEZ, L. A., COLPAS, F. J., AND MÜLLER, E. A. A Simplified Cubic-Equation-of-State Correlation for Solubility of Solids in Supercritical Fluids, paper presented at 1994 National Meeting, Atlanta, GA. April 17-21, 1994.
- [25] EWALD, A. K., JEPSON, W. B., AND ROWLINSON, J. S. The Solubility of Solids in Gases. *Disc. Farad. Soc.* 15 (1953), 238-243.
- [26] FINCK, U., WAOHLFARTH, C., AND HEUER, T. *Ber. Bunsenges. Phys. Chem.* 96 (1992), 179.
- [27] HASELOW, J. S., HAN, S. J., GREENKORN, R. A., AND CHAO, K. C. Equations of State for Supercritical Extraction. No. 300 in ACS Symp. Ser., pp. 156-178. American Chemical Society, Washington, DC, 1986.
- [28] HAWTHORNE, S. B., AND MILLER, D. J. Extraction and Recovery of Polycyclic Aromatic Hydrocarbons from Environmental Solids Using Supercritical Fluids. *Anal. Chem.* 59 (1987), 1705-1708.
- [29] HINCKLEY, R. B., AND REID, R. C. *AIChE J.* 10 (1964), 416.

- [30] IWAI, Y., FUKUDA, T., KOGA, Y., AND ARAI, Y. Solubilities of Myristic Acid, Palmitic Acid and Cetyl Alcohol in Supercritical Carbon Dioxide at 35 °C . *J. Chem. Eng. Data* 36 (1991), 430-432.
- [31] IWAI, Y., KOGA, Y., FUKUDA, T., AND ARAI, Y. Correlation of Solubilities of High-Boiling Components in Supercritical Carbon Dioxide Using a Solution Model . *J. Chem. Eng. Jap.* 25 (1992), 757-760.
- [32] JOHNSTON, K. P., AND PECK, D. G. Modeling Supercritical Mixtures. How Predictive Is It ? *Ind. Eng. Chem. Res.* 28 (1989), 1115-1125.
- [33] JOHNSTON, K. P., PECK, D. G., AND KIM, S. Modeling Supercritical Mixtures: How Predictive is it ? . *Ind. Eng. Chem. Res.* 28 (1989), 1115-1125.
- [34] JOHNSTON, K. P., AND PENNINGER, J. M. L., Eds. *Supercritical Fluid Science and Technology* , no. 406 in ACS Symp. Ser., American Chemical Society, Washington DC, 1989.
- [35] JOHNSTON, K. P., ZIGER, D. H., AND ECKERT, C. A. Solubilities of Hydrocarbon Solids in Supercritical Fluids. The Augmented van der Waals Treatment. *Ind. Eng. Chem. Fundam.* 21 (1982), 191-197.
- [36] KIM, S., AND JOHNSTON, K. P. Clustering in Supercritical Fluid Mixtures. *AIChE J.* 33 (1987), 1603-1611.
- [37] KNEZ, Z., AND STEINER, R. J. *J. Supercrit. Fluids* 5 (1992), 251.
- [38] KOHN, P. M., AND SAVAGE, P. R. Supercritical Fluids Try for CPI Applications. *Chem. Eng.* 86, 6 (1979), 41-43.
- [39] KOSAL, E., LEE, C. H., AND HOLDER, G. D. *J. Supercrit. Fluids* 5 (1992), 169.
- [40] KRAMER, A., AND THODOS, G. Solubility of 1-Hexadecanol and Palmitic Acid in Supercritical Carbon Dioxide . *J. Chem. Eng. Data* 33 (1988), 230-234.
- [41] LARSEN, B. L., RAMMUSSEN, P., AND FREDENSLUND, A. A Modified UNI-FAC Group-Contribution Model for Prediction of Phase Equilibria and Heats of Mixing. *Ind. Eng. Chem. Res.* 26 (1987), 2274-2286.

- [42] LARSON, K. A., AND KING, M. L. Evaluation of Supercritical Fluid Extraction in the Pharmaceutical Industry. *Biotechnol. Prog.* 2 (1986), 73-82.
- [43] LAVAL, A. S., AND VAN DER LAAN, E. T. . A Partial Molar Fugacity Coefficient Useful for Changing Mixing Rules of Cubic Equations of State. *Fluid Phase Equilibria* 95 (1994), 109-121.
- [44] LEE, Y. Y., KIM, H., LEE, H., AND HONG, W. H. *Korean J. Chem. Eng.* 5 (1988), 76.
- [45] LIDE, D. R. *Handbook of Chemistry and Physics*. 71st ed. CRC Press, Boca Raton, 1990.
- [46] MAGOULAS, K., AND TASSIOS, D. Thermophysical Properties of n-Alkanes C_1 to C_{20} and their Prediction for Higher Ones . *Fluid Phase Equilibria* 56 (1990), 119-140.
- [47] MCHUGH, M., AND KRUKONIS, V. *Supercritical Fluid Extraction. Principles and Practice* , 1st ed. Butterworth-Heinemann, MA, 1985.
- [48] MCHUGH, M., AND KRUKONIS, V. *Supercritical Fluid Extraction. Principles and Practice* , 2nd ed. Butterworth-Heinemann, Stoneham, MA, 1994.
- [49] MIKA, V. A Generalized Treatment of Cubic Equations of State. *Collect. Czech. Chem. Commun.* 54 (1989), 2879-2895.
- [50] MUKHOPADHYAY, M., AND RAO, G. V. R. Thermodynamic Modeling for Supercritical Fluid Process Design. *Ind. Eng. Chem. Res.* 32 (1993), 922-929.
- [51] PARKINSON, G., AND JOHNSON, E. Supercritical Processes Win CPI acceptance. *Chem. Eng.* 96, 7 (1989), 35-39.
- [52] PAULAITIS, M. E., KRUKONIS, V. J., KURNIK, R. J., AND REID, R. C. Supercritical Fluid Extraction. *Rev. Chem. Eng.* 1 (1983), 179.
- [53] PAULAITIS, M. E., KRUKONIS, V. J., KURNIK, R. J., AND REID, R. C. The Use of Entrainers in the Supercritical Extraction of Soils Contaminated with Hazardous Organics. *Ind. Eng. Chem. Res.* 26 (1987), 2058-2062.
- [54] PENG, D. Y., AND ROBINSON, D. B. A New Two-Constant Equation of State . *Ind. Eng. Chem. Fundam.* 15 (1976), 59-66.

- [55] POLITZER, P., MURRAY, J. S., LANE, P., AND BRINCK, T. J. Relationships between Solute Molecular Properties and Solubility in Supercritical CO₂. *J. Phys. Chem.* 97 (1993), 729-732.
- [56] PONGSIRI, N., AND VISWANATH, D. S. A Comparative Study of Mixing Rules in the Prediction of Solid-Vapor Equilibria. *Ind. Eng. Chem. Res.* 28 (1989), 1918-1921.
- [57] PRAUSNITZ, J. M., LICHTENTHALER, R. N., AND DE AZEVEDO, E. G. *Molecular Thermodynamics Of Fluid-Phase Equilibria*. Prentice-Hall, Englewood Cliffs, NJ, 1986.
- [58] QUIRAM, D. J., ÓCONNELL, J. P., AND COCHRAN, H. D. The Solubility of Solids in Compressed Gases. *J. Supercrit. Fluids* 7 (1994), 159-164.
- [59] RAO, V. S. G., AND MUKHOPADHYAY, M. Proc. Int. Symp. Supercritical Fluids. p. 161, 1988.
- [60] REDLICH, O., AND KWONG, J. N. S. On the Thermodynamics of Solutions. V. An Equation of State. Fugacities of Gaseous Solutions. *Chem. Rev.* 44 (1949), 233-244.
- [61] REID, R. C., PRAUSNITZ, J. M., AND POLING, B. E. *The Properties of Gases and Liquids*, 4th ed. McGraw-Hill, New York, 1987.
- [62] ROBIN, S., AND VODAR, B. Solubility in Compressed Gases. *Disc. Faraday Soc.* 15 (1953), 233-238.
- [63] ROOP, R. K., HESS, K. R., AND AKGERMAN, A. Supercritical Extraction of Pollutants from Water and Soil. In *Supercritical Fluid Science and Technology*, K. P. Johnston and J. M. L. Penninger, Eds., no. 406 in ACS Symp. Ser., pp. 468-476, American Chemical Society, Washington DC, 1989.
- [64] SAKAKI, K. Solubility of β -Carotene in Dense Carbon Dioxide and Nitrous Oxide from 308 to 320 K and from 9.6 to 30 MPa. *J. Chem. Eng. Data* 37 (1992), 249-251.
- [65] SCHMIDT, G., AND WENZEL, H. A Modified van der Waals Type Equation of State. *Chem. Eng. Sci.* 35 (1980), 1503-1512.

- [66] SCHMITT, W. J., AND REID, R. C. In *Supercritical Fluid Technology*. J. M. L. Penninger, M. Radosz, M. A. Hugh and V. J. Krukonis, Eds., p. 123. Elsevier, Amsterdam, 1985.
- [67] SCHMITT, W. J., AND REID, R. C. Solubility of Monofunctional Organic Solids in Chemically Diverse Supercritical Fluids. *J. Chem. Eng. Data* 31 (1986), 204-212.
- [68] SHENG, Y. J., CHEN, P. C., CHEN, Y. P., AND WONG, D. S. H. Calculation of Solubilities of Aromatic Compounds in Supercritical Carbon Dioxide. *Ind. Eng. Chem. Res.* 31 (1992), 967-973.
- [69] SHENG, Y. J., CHEN, Y. P., AND WONG, D. S. H. A Cubic Equation of State for Predicting Vapor-Liquid Equilibria of Hydrocarbon Mixtures Using a Group Contribution Mixing Rule. *Fluid Phase Equilibria* 46 (1989), 197-210.
- [70] SOAVE, G. Equilibrium Constants from a Modified Redlich-Kwong Equation of State. *Chem. Eng. Sci.* 27 (1972), 1197-1203.
- [71] SOAVE, G. Correlation of the Solubility of Ill-Characterized Solid Compounds in Supercritical Fluids by a Cubic Equation of State. *Fluid Phase Equilibria* 64 (1991), 165-171.
- [72] SPILIOTIS, N., MAGOULAS, K., AND TASSIOS, D. Prediction of the Solubility of Aromatic Hydrocarbons in Supercritical CO₂ with EOS/G^E Models. *Fluid Phase Equilibria* 102 (1994), 121-141.
- [73] SZARAWARA, J., AND GAWDZIK, A. Method of Calculation of Fugacity Coefficient from Cubic Equations of State. *Chem. Eng. Sci.* 44 (1989), 1489-1494.
- [74] TODD, D., AND ELGIN, J. Phase Equilibria in Systems With Ethylene Above its Critical Temperature. *AIChE J.* 1 (1955), 20-?
- [75] WANG, X., AND TAVLARIDES, L. L. Solubility of Solutes in Compressed Gases. *Ind. Eng. Chem. Res.* 33 (1994), 724-729.
- [76] WENG, W. L., AND LEE, M. J. Phase Equilibrium Measurements for Binary Mixtures of Methyl Benzoate Plus CO₂, C₂H₆ and C₂H₄. *J. Chem. Eng. Jap.* 25 (1992), 211-215.

- [77] YU, E., RICHTER, M., CHEN, P., WANG, X., ZHANG, Z., AND TAVLARIDES, L. L. Solubilities of Polychlorinated Biphenyls in Supercritical Carbon Dioxide. *Ind. Eng. Chem. Res.* 34 (1995), 340-346.

NOTE ADDED IN PROOF

Appendix A

Program Listing of the Rigorous Model

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "rosen.h"

#define R 83.14
#define ITER 100
#define MAX_ITER 20
#define TOL 1.0e-5
#define TOLRNS 1.0e-10
#define N 20
#define Size 100
#define sq(x) ((x)*(x))

/*...Input Variables...*/

char *fn1="../data/ethyln", *fn2="../data/2_3dmn", *dat,
*Pydat = "../data/c2h4_3d2" ;
int opt,mr,eos;
int i,u,w,N_exp;
double mi,tc1,pc1,accentric1;
double mj,tc2,pc2,accentric2,pA2,pB2,pC2,Vs,p_vap;
double t,P[N],y2_exp[N];
double omega_a,omega_b;
double CONVRG = 0.0 ;

char *fn[]={
"vdW",
```

```
"SRK",
"PR",
"RK"};
FILE *fp,*fq,*fr,*fd,*sol_fp;

/*...Calculated Variables...*/

double a1,a2,b1,b2,B_1,B_2;

/* solmain.c */

#include <stdio.h>
#include <math.h>
#include "rosen.h"
#include "solblty.c"
#include "rosen.c"
#include "sol_input.c"
#include "eql.c"
#include "prdeql.c"

extern double avg_error(double) ;
char *sol_fn="sol.out";

void main()
{
int i,k=0,mr_max=2 ;
double dif, pred_value ;
extern int no_variables;
int variable;
extern char *dat;
extern double (*rosenbrock)(),temp_vector[][MAX_VARIABLE];
double Avg_Err[32] ;
double (*fp_prdeql)() ;
char *mr_name ;
char *prsnt_mr[]={
"QMR_&_QMR",
"CMR_&_QMR",
"QMR_&_LMR",
"CMR_&_LMR",
"Wong_&_Sand" };

char *table_fn[] ={
"1p_QMR",
"1p_CMR",
"2p_QMR",
"2p_CMR",
"1p_QLMR",
"1p_CLMR"   };

FILE *res1_fp, *res2_fp, *table_fp[7] ;
```



```

if ( (sol_fp=fopen(sol_fn,"at")) == NULL ){
fprintf(stderr,"Error opening file: %s\n",
    sol_fn);
exit(1);
}

for(i=0;i<=3;i++) {
if ( (table_fp[i]=fopen(table_fn[i],"at")) ==
    NULL ){
fprintf(stderr,"Error opening file: %s\n",
    table_fp[i]);
exit(1);
}
}

tpy_input() ;
system_input();

fprintf(sol_fp,"\n\t*** S Y S T E M ***\n");
fprintf(sol_fp,"\n%s*\t*%s*\t*%s*\n",fn1,fn2,Pydat);
fprintf(sol_fp,"\nTemperature : %lf\tNo. of Data points : %d\n",
    t,N_exp);

re:
printf("\nEnter the no of Variables :");
scanf("%d",&no_variables);

if (no_variables){
/*if (no_variables == 2)
mr_max = 2 ;*/
fprintf(sol_fp,"\n\nThe no. of OPTIMIZATION parameters : %d",
    no_variables);
for(mr=1;mr<=mr_max;mr++) {
mr_name = prsnt_mr[--mr];
mr++;
if (no_variables == 1) {
fprintf(table_fp[mr-1],
    "\n%s\t%s\t%.2lf\t%d\t",
    fn2,fn1,t,N_exp) ;
}
else {
fprintf(table_fp[mr+1],
    "\n%s\t%s\t%.2lf\t%d\t",
    fn2,fn1,t,N_exp) ;
}
}
fprintf(sol_fp,"\n\nMixing Rule for \"a\" & \"b\" is %s",
    mr_name);
for(eos=3;eos<=3;eos++) {
Select_EoS(eos);
Calc_Constants(eos);
rosen();
pred_value = avg_error(((*)

```

```
    rosenbrock)(temp_vector));

pred_value = predict_eq1(mr,
    eos,temp_vector) ;

fprintf(sol_fp,"\n%s---",
    dat);

if (no_variables == 1) {
fprintf(sol_fp,
    "\t kij = %.4lf\tAvg_Err = %.2lf ",
    temp_vector[0][0],
    pred_value);
fprintf(table_fp[mr-1],
    "%.4lf\t%.2lf\t",
    temp_vector[0][0],
    pred_value) ;
}

else {
fprintf(sol_fp,
    "\tkij = %.4lf\tlij = %.4lf\tAvg_Err = %.2lf",
    temp_vector[0][0],
    temp_vector[0][1],
    pred_value);
fprintf(table_fp[mr+1],
    "%.4lf\t%.4lf\t%.2lf\t",
    temp_vector[0][0],
    temp_vector[0][1],
    pred_value) ;
}

/* eq1(mr,eos,temp_vector) ; */

printf("\n**AVG. ERROR : %lf\n",
    pred_value);
}

if (no_variables == 1) {
fclose(table_fp[mr-1]) ;
}
else {
fclose(table_fp[mr+1]) ;
}
}

goto re;
}

}
```

```
/* solblty.c */
```

```
#include "solblty.h"
```

```
#include "mix_rule.c"
#include "fug_SW_EoS.c"
#include "fug_PR_WS.c"

/* Calculation of a,b */
void calc_const(eos,tc,pc,accentric,t,omega_a,omega_b,a,
b)
int eos;
double tc,pc,accentric,t,omega_a,omega_b,*a,*b;
{
double tr;
double alpha,beta;
tr = t/tc;
switch(eos)
{
case 1:
*a = omega_a * R*R * tc*tc / pc ;
*b = omega_b * R * tc / pc ;
break;

case 2:
*a = omega_a * R*R * pow(tc,2.0) / pc ;
beta = 0.480 + 1.574*accentric - 0.176*
accentric*accentric ;
alpha = 1.0 + beta * (1.0 - sqrt(tr)) ;
alpha *= alpha;
*b = omega_b * R * tc / pc ;
break;

case 3:
*a = omega_a * R*R * tc*tc / pc ;
beta = 0.37464 + 1.54226*accentric - 0.26992*
accentric*accentric ;
alpha = 1.0 + beta * (1.0 - sqrt(tr)) ;
alpha *= alpha;
*a *= alpha;
*b = omega_b * R * tc / pc ;
break;

case 4:
*a = omega_a * R*R * pow(tc,2.5) / (pc*
sqrt(t)) ;
*b = omega_b * R * tc / pc ;
break;
default :
break;
}
}

/* Schmitt-Wenzel EOS in 'z' form */
```

```

double f(A,B,z,u,w)
int u,w;
double A,B,z;
{
return z*z*z + (u*B-B-1.0)*z*z + (w*B*B - u*B*B
    - u*B + A)*z - (A*B + w*B*B +w*B*B*B);
}

/* Differential of S-W-Eos */

double df(A,B,z,u,w)
int u,w;
double A,B,z;
{
return 3.0*z*z + 2.0*(u*B - B - 1.0)*z + (w*B*
    B - u*B*B - u*B + A);
}

/* Calculation "vap-z" & vap-vol. */

double calc_v(u,w,p,t,a,b,guess)
int u,w;
double p,t,a,b,guess;
{
double z,diff,dfz,fz,v;
double A,B;
int i=0;

B = b*p / (R*t);
A = a*p / (R*R*t*t);
z = guess;
do
{
dfz = df(A,B,z,u,w);
/*if (dfz<TOL)
{
fprintf(stderr,"Denom < TOL in newton raphson for Z");
exit(3);
}*/
fz = f(A,B,z,u,w);
guess = z - fz/dfz;
diff = fabs(z -guess);
z = guess;
}while(i++<ITER && diff>TOL);
if (diff>TOL) /*not converged*/
return -100;
/*printf("\nz=%lf",z);*/
v = R*t*z/p;
return v;
}

double solblty(t,p,fug2)
double t,p,fug2 ;

```

```

{
double pf;
pf= exp(Vs*(p-p_vap)/(R*t));

return p_vap*pf/(p*fug2);
}

double func(x)
double x[][MAX_VARIABLE];
{
int i,j, stat;
double y2_guess, dif;
double guess=1.0,y2_cal[N],diff[N],sum=0.0;
double v_by_SW,am,bm,fug2_SW,OF,k,l,fug2_PR;
double Q,D,d_D,d_Q,gamma_i,Ae ;

B_1 = b1 - a1/(R*t) ;
B_2 = b2 - a2/(R*t) ;
k = x[0][0] ;
if (no_variables == 1)
l =0.0 ;
else
l = x[0][1] ;
for (i=0;i<N_exp;i++) {
j =0 ;
y2_cal[i] = y2_exp[i];
do {
y2_guess= y2_cal[i];
if (mr == 5) {
eval_Ae(y2_guess,P[i],&
Ae,&gamma_i) ;
eval_D_Q_wsmr(y2_guess,
k,Ae,gamma_i,&Q,&D,&d_Q,
&d_D);
}
mr_for_a_b(y2_guess,k,l,Q,D,&am,
&bm);
v_by_SW = calc_v(u,w,P[i],t,am,
bm,guess);
/* if (!j && v_by_SW <= bm ) {
printf("\n bm is greater than v !!!!!") ;break;} */

fug2_SW = eval_fug_SW(k,l,u,w,P[i],
v_by_SW,t,am,bm,D,Q,d_D,d_Q,y2_guess);
fug2_PR = eval_fug_PR_WS(y2_guess,
P[i],v_by_SW,t,am,bm,Ae,gamma_i,
k) ;
y2_cal[i] = solblty(t,P[i],fug2_SW);
dif = fabs(y2_guess-y2_cal[i]) ;
} while (j++ < MAX_ITER && dif > TOL) ;

}
for (i=0;i<N_exp;i++)
{

```

```
diff[i] = (y2_cal[i] - y2_exp[i])*(y2_cal[i]
    - y2_exp[i]);
sum += diff[i] ;
}
OF = sqrt(sum / N_exp) ;
return OF ;
}
```

```
double avg_error(OF)
double OF;
{
    int i;
    double sum=0.0;
    for (i=0;i<N_exp;i++)
    {
        sum += y2_exp[i];
    }

    return OF*N_exp*100.0/sum ;
}
```

```
void Select_EoS(eos)
{
    switch(eos)
    {
        case 1:
            dat=fn[0];
            break;
        case 2:
            dat=fn[1];
            break;
        case 3:
            dat=fn[2];
            break;
        case 4:
            dat=fn[3];
            break;
        default:
            break;
    }

    if ( (fp=fopen(dat,"rt")) == NULL )
    {
        fprintf(stderr,"fopen failure : %s\n",dat);
        exit(1);
    }
    fscanf(fp,"%d%d%lf%lf",&u,&w,&omega_a,&omega_b);
    fclose(fp);
}
```

```
void Calc_Constants(eos)
{
    calc_const(eos,tc1,pc1,accentric1,t,omega_a,omega_b,
        &a1,&b1);
    calc_const(eos,tc2,pc2,accentric2,t,omega_a,omega_b,
        &a2,&b2);
}
```

```
/* fug_SW_EoS.c */
```

```
/* Determinent */
```

```
double eval_D(u,w)
int    u,w;
{
    return( u*u-4*w );
}
```

```
/* Roots */
```

```
void eval_r1_r2(u,w,bm,r1,r2)
int u,w;
double bm,*r1,*r2;
{
    double D;

    D = eval_D(u,w);
    if (D<0.)
    {
        printf("Delta is negative!");
        exit(1);
    }
    *r1 = ((-u+sqrt(D))/2)*bm;
    *r2 = ((-u-sqrt(D))/2)*bm;
}
```

```
/* First term in fugacity expression */
```

```
double term_01(p,v,t)
double p,v,t;
{
    return( p*v/(R*t) -1.0 );
}
```

```
/* Second term in fugacity expression */
```

```
double term_02(t,p)
double t,p;
{
```

```
double vd;
vd = R*t/p;
return( log(vd) );
}

/* Third term in fugacity expression */

double term_03(u,w,v,bm,t,am)
int u,w;
double v,bm,t,am;
{
double temp,D,r1,r2;
D=eval_D(u,w);
eval_r1_r2(u,w,bm,&r1,&r2);
if (D==0.0)
{
return ( -log(v-bm) - am/(R*t*(v-r1)) ) ;
}
if (D>0.0)
{
temp = am * log((v-r2)/(v-r1)) / (R*t*(r1-
r2));
return (-log(v-bm) - temp);
}
}

/* Fourth term in fugacity expression */

double term_04(u,w,v,bm)
double v,bm;
int u,w;
{
double D,r1,r2;
D= eval_D(u,w);
eval_r1_r2(u,w,bm,&r1,&r2);
if (D>0.)
{
return ( -log((v-r2)/(v-r1)) / (r1-r2) );
}
if (D==0.)
{
return (-1)/(v-r1);
}
}

/* Fifth term in fugacity expression */

double term_05(u,w,a,b,v)
double b,a,v;
int u,w;
{
double D,r1,r2;
double t1,t2,t2a,t3a,t3,t4;
D=eval_D(u,w);
```



```
eval_r1_r2(u,w,b,&r1,&r2);

if (D==0.)
{
t1= (R*t)/(v-b);
t2= a*u/(2*pow(v-r1,2.0));
t3= (2*w*a*b + r1*a*u)/(3*pow(v-r1,3.0));
return t1+t2+t3 ;
}
if (D>0.)
{
t1= (R*t)/(v-b);
t2a= v*v + u*v*b + w*b*b;
t2= a*u*0.5/t2a;
t3a= (2*w*a*b-a*u*u*b*0.5 )/((r1-r2)*(r1-
r2));
t3= t3a*(2.0*v - (r1+r2))/t2a;
t4= t3a*2.0*log((v-r2)/(v-r1))/(r1-r2);
return t1+t2+t3-t4;
}
}

/* Fifth (a) in fugacity expression when LMR for "b" */

double term_05_a_lmr(bi)
double bi;
{
return (bi);
}

/* Fifth (b) in fugacity expression when LMR for "b' */

double term_05_b_lmr(bm)
double bm;
{
return bm ;
}

/* Fifth (a) in fugacity expression when QMR for "b" */

double term_05_a_qmr(bi,bm,lij)
double bi,bm,lij;
{
return (bi+bm)*(1.0 - lij);
}

/* Fifth (b) in fugacity expression when QMR for "b' */

double term_05_b_qmr(bm)
double bm;
{
return( 2.0*bm );
}
```

```

}

/* Fourth (b) in fugacity expression when QMR for "a" */

double term_04_b_qmr(am)
double am;
{
return( 2.0*am );
}

/* Fourth (a) in fugacity expression when QMR for "a" */

double term_04_a_qmr(ai,am,kij)
double ai,am,kij;
{
return 2.0*sqrt(ai*am)*(1.0 - kij);
}

/* Fourth (a) in fugacity expression when CMR for "a" */

double term_04_a_cmr(kij,x2,am)
double kij,x2,am;
{
double x1,t1,t2,t3,t4;
x1 = 1.0 - x2;
t1= pow(mi,.3333)*pow(a2*mj,.3333);
t2= pow(mj,.3333)*pow(a1*mi,.3333);
t3= sq(x1*pow(mi,.3333) + x2*pow(mj,.3333));
t4= pow(sqrt(a1*a2),.3333) * (1-kij) * (1 - 2*x2);
return( 3*pow(am,.6667)* ( ((t1-t2)/t3) +t4) );
}

/* Fourth (b) in fugacity expression when CMR for "a" */

double term_04_b_cmr(kij,x2,am)
double kij,x2,am;
{
double x1,t1,t2,t3,t4;
x1 = 1.0 - x2;
t1 = pow(mi,.3333)*pow(a2*mj,.3333);
t2=pow(mj,.3333)*pow(a1*mi,.3333);
t3=sq(x1*pow(mi,.3333)+x2*pow(mj,.3333));
t4=pow(sqrt(a1*a2),.3333)*(1-kij)*(1-2*(x2*x2+x1*
x1));
return( 3*pow(am,.6667) * ( ((x1-x2)*(t1-t2)/t3) +
t4) );
}

/* Fourth (a)&(b) in fugacity expression when WS-MR */
void term_04_a_b_wsmr(bm,D,d_D,t5a,t5b,t4a,t4b)
double bm,D,d_D,t5a,t5b,*t4a,*t4b ;
{
*t4b = R * t * ( bm * D + D * t5b ) ;

```

```
*t4a = R * t * ( bm * d_D + D * t5a ) ;
return ;
}

/*
void term_04_a_b_wsmr(D,Q,d_D,d_Q,t4a,t4b)
double D,Q,d_D,d_Q,*t4a,*t4b ;
{
    double temp1,temp2 ;

    temp1 = Q/sq(1.0 - D) ; temp2 = D/(1.0 - D) ;

    *t4b = R * t * ( D * temp1 - 2 * Q * temp2 ) ;
    *t4a = R * t * ( temp1 * d_D + temp2 * d_Q ) ;
}
*/

/* Fifth (a)&(b) in fugacity expression when WS-MR */

void term_05_a_b_wsmr(D,Q,d_D,d_Q,t5a,t5b)
double D,Q,d_D,d_Q,*t5a,*t5b ;
{
    double temp1,temp2 ;

    temp1 = Q/sq(1.0 - D) ;
    temp2 = D/(1.0 - D) ;

    *t5b = D * temp1 + 2 * temp2 * (Q/D) ;
    *t5a = temp1 * d_D + d_Q / (1.0 - D) ;
}

void term_04_a_b(mr,ai,am,x2,kij,t4a,t4b)
int mr;
double ai,am,x2,kij,*t4a,*t4b ;
{
    switch(mr)
    {
        case 1 :
            *t4a = term_04_a_qmr(ai,am,kij);
            *t4b = term_04_b_qmr(am);
            break;
        case 3 :
            *t4a = term_04_a_qmr(ai,am,kij);
            *t4b = term_04_b_qmr(am);
            break;
        case 2 :
            *t4a = term_04_a_cmr(kij,x2,am);
            *t4b = term_04_b_cmr(kij,x2,am);
            break;
        case 4 :
            *t4a = term_04_a_cmr(kij,x2,am);
            *t4b = term_04_b_cmr(kij,x2,am);
            break;
        default :
```

```

break;
}
}

void term_05_a_b(mr,bi,bm,lij,t5a,t5b)
int mr;
double bi,bm,lij,*t5a,*t5b;
{
switch(mr)
{
case 1 :
*t5a = term_05_a_qmr(bi,bm,lij);
*t5b = term_05_b_qmr(bm);
break;
case 3 :
*t5a = term_05_a_lmr(bi);
*t5b = term_05_b_lmr(bm);
break;
case 2 :
*t5a = term_05_a_qmr(bi,bm,lij);
*t5b = term_05_b_qmr(bm);
break;
case 4 :
*t5a = term_05_a_lmr(bi);
*t5b = term_05_b_lmr(bm);
break;
default :
break;
}
}

/* Evaluate fugacity of a component */

double eval_fug_SW(k,l,u,w,p,v,t,am,bm,D,Q,d_D,d_Q,x2)
int u,w;
double p,v,t,am,bm,k,l,x2,D,Q,d_D,d_Q;
{
double t1,t2,t3,t4,t5,t4a,t4b,t5a,t5b;

t1    = term_01(p,v,t) ;
t2    = term_02(t,p) ;
t3    = term_03(u,w,v,bm,t,am) ;
if (mr==5) {
term_05_a_b_wsmr(D,Q,d_D,d_Q,&t5a,&t5b) ;
term_04_a_b_wsmr(bm,D,d_D,t5a,t5b,&t4a,
&t4b) ;
/*
term_04_a_b_wsmr(D,Q,d_D,d_Q,&t4a,&t4b) ;
*/
}
else {
term_04_a_b(mr,a2,am,x2,k,&t4a,&t4b);
term_05_a_b(mr,b2,bm,l,&t5a,&t5b);

```

```
}
t4   = term_04(u,w,v,bm);
t5   = term_05(u,w,am,bm,v);

return exp( t1+t2+t3+(t4a-t4b)*t4/(R*t)+(t5a-t5b)*
            t5/(R*t) ) ;
}

/* mix_rule.c */

/* Fifth (a) in fugacity expression when LMR for "b" */

double term_05_a_lmr(bi)
double bi;
{
return (bi);
}

/* Fifth (b) in fugacity expression when LMR for "b' */

double term_05_b_lmr(bm)
double bm;
{
return bm ;
}

/* Fifth (a) in fugacity expression when QMR for "b" */

double term_05_a_qmr(bi,bm,lij)
double bi,bm,lij;
{
return (bi+bm)*(1-lij);
}

/* Fifth (b) in fugacity expression when QMR for "b' */

double term_05_b_qmr(bm)
double bm;
{
return( 2.0*bm );
}

/* Fourth (b) in fugacity expression when QMR for "a" */

double term_04_b_qmr(am)
double am;
{
return( 2.0*am );
}

/* Fourth (a) in fugacity expression when QMR for "a" */
```

```
double term_04_a_qmr(ai,am,kij)
double ai,am,kij;
{
return 2.0*sqrt(ai*am)*(1.0 - kij);
}
```

```
/* Fourth (a) in fugacity expression when CMR for "a" */
```

```
double term_04_a_cmr(kij,x2,am)
double kij,x2,am;
{
double x1,t1,t2,t3,t4;
x1 = 1.0 - x2;
t1= pow(mi,.3333)*pow(a2*mj,.3333);
t2= pow(mj,.3333)*pow(a1*mi,.3333);
t3= sq(x1*pow(mi,.3333) + x2*pow(mj,.3333));
t4= pow(sqrt(a1*a2),.3333) * (1-kij) * (1 - 2*x2);
return( 3*pow(am,.6667)* ( ((t1-t2)/t3) +t4) );
}
```

```
/* Fourth (b) in fugacity expression when CMR for "a" */
```

```
double term_04_b_cmr(kij,x2,am)
double kij,x2,am;
{
double x1,t1,t2,t3,t4;
x1 = 1.0 - x2;
t1=pow(mi,.3333)*pow(a2*mj,.3333);
t2=pow(mj,.3333)*pow(a1*mi,.3333);
t3=sq(x1*pow(mi,.3333)+x2*pow(mj,.3333));
t4=pow(sqrt(a1*a2),.3333)*(1-kij)*(1-2*(x2*x2+x1*
x1));
return( 3*pow(am,.6667) * ( ((x1-x2)*(t1-t2)/t3) +
t4) );
}
```

```
/* prdeql.c */
```

```
/*.. Predict Equilibria ..*/
```

```
extern double avgerror(double) ;
```

```
double predict_eq1(int opt_mr, int opt_eos, double x[][MAX_VARIABLE])
{
int i,j;
double y2_guess, dif;
double guess=1.0,y2[N],diff[N],sum=0.0;
double v_by_SW,am,bm,fug2_SW,OF,k,l,fug2_PR;
double Q,D,d_D,d_Q,gamma_i,Ae ;
double T_pred, P_exp[N] ;
char Pred_Pydat[20] , *eq1 = "eq1.out";
```

```

FILE *pred_fp,*eql_fp ;

/*retry :fprintf(stderr,"\nEnter *T-P-y* Filename : ");
    scanf("%s",Pred_Pydat);*/
if ( (pred_fp=fopen(Pydat,"rt")) == NULL )
{
fprintf(stderr,"fopen failure : %s\n",Pydat);
}
fscanf(pred_fp,"%lf%d",&T_pred,&N_exp);
for(i=0;i< N_exp;i++)
{
fscanf(pred_fp,"%lf%lf",&P_exp[i],&y2_exp[i]);
}
fclose(pred_fp);
fprintf(sol_fp,"\n\n** Prediction at %.2lfK **",
    T_pred);
mr = opt_mr ;
eos = opt_eos ;
Select_EoS(eos) ;
Calc_Constants(eos) ;

B_1 = b1 - a1/(R*t) ;
B_2 = b2 - a2/(R*t) ;

k = x[0][0] ;
if (no_variables == 1)
l = 0.0 ;
else
l = x[0][1] ;

k = 0.0;
l = 0.0;

for (i=0;i<N_exp;i++) {
j = 0 ;
y2[i] = y2_exp[i];
do {
y2_guess= y2[i];
if (mr == 5) {
eval_Ae(y2_guess,P_exp[i],
    &Ae,&gamma_i) ;
eval_D_Q_wsmr(y2_guess,
    k,Ae,gamma_i,&Q,&D,&d_Q,
    &d_D);
}
mr_for_a_b(y2_guess,k,l,Q,D,&am,
    &bm);
v_by_SW = calc_v(u,w,P_exp[i],T_pred,
    am,bm,guess);
/* if (!j && v_by_SW <= bm ) {
    printf("\n bm is greater than v !!!!!") ;break;} */

```

```

fug2_SW = eval_fug_SW(k,l,u,w,P_exp[i],
    v_by_SW,T_pred,am,bm,D,Q,d_D,d_Q,
    y2_guess);
fug2_PR = eval_fug_PR_WS(y2_guess,
    P_exp[i],v_by_SW,T_pred,am,bm,Ae,
    gamma_i,k) ;
y2[i] = solblty(T_pred,P_exp[i],
    fug2_SW);
dif = fabs(y2_guess-y2[i]) ;
} while ( dif > TOLRNS) ;
}
for (i=0;i<N_exp;i++)
{
diff[i] = (y2[i] - y2_exp[i])*(y2[i] -
    y2_exp[i]);
sum += diff[i] ;
}
OF = avg_error(sqrt(sum / N_exp) );
if (!avg_error) {
printf("\n Avg. Error is ZERO , Can you believe it !!") ;
exit(1);
}
printf("\n\nAvg. Error of Prediction : %lf",OF) ;
printf("\n\nCalculated \t Experimental") ;
for (i=0;i<N_exp;i++) {
printf("\n%e \t %e",y2[i],y2_exp[i]) ;
}
fprintf(sol_fp,"\n\nCalculated \t Experimental") ;
fprintf(sol_fp,"\n_____");
for (i=0;i<N_exp;i++) {
fprintf(sol_fp,"\n%e \t %e",y2[i],y2_exp[i]) ;
}
fprintf(sol_fp,"\n\nAvg. Error of Prediction : %lf",
    OF) ;
fprintf(sol_fp,"\n*****\n") ;
return OF ;
}

/* rosen.h */

/*..
    Constant Declarations ../
#define COUNTER 10
#define TOLERANCE .0001
#define MAX_FUNCTION 10
#define MAX_VARIABLE 10

/*..
    Global Declarations ../
double x[MAX_VARIABLE][MAX_VARIABLE];
double (*function[MAX_FUNCTION])();          /* function pointer */
int dimension[MAX_VARIABLE],no_test_function,times_function_calculated;

```



```
int no_variables;
FILE *out;

double (*rosenbrock)(),temp_vector[1][MAX_VARIABLE];

int mr,eos;

/*    rosen.c    */

#include <stdio.h>
#include <math.h>
#include "rosen.h"

/* extern double opt_paramtrs[][MAX_VARIABLE] ;*/

void initiate()
{
/*..
    Total numbers of test function written here ..*/
no_test_function=2;
/*..
    "func" is the base address of objective the function ..*/
function[1]=func;
dimension[1]=no_variables;
function[2]=ycal;
dimension[2]=no_variables ;
}

void sum_matrix(m,l,a,b,c)
int m,l;
double a[][MAX_VARIABLE],b[][MAX_VARIABLE],c[][MAX_VARIABLE];
{
int i,j;
for(i=0 ; i<m ; ++i)
for(j=0 ; j<l ; ++j)
c[i][j]=a[i][j]+b[i][j];
return;
}

void mult_matrix(m,p,l,a,b,c)
int m,p,l;
double a[][MAX_VARIABLE],b[][MAX_VARIABLE],c[][MAX_VARIABLE];
{
int i,j,k;
double sum;
for(i=0 ; i<m ; ++i)
for(j=0 ; j<l ; ++j)
{
sum=0.0;
```

```

for(k=0 ; k<p ; ++k)
sum+=a[i][k]*b[k][j];
c[i][j]=sum;
}
return;
}

```

```

void scalar_mult(m,l,a,b,constant)
int m,l;
double a[][MAX_VARIABLE],b[][MAX_VARIABLE],constant;
{
int i,j;
for(i=0 ; i<m ; ++i)
for(j=0 ; j<l ; ++j)
b[i][j]=a[i][j]*constant;
return;
}

```

/****** MAIN PROGRAM *****/

```

void rosen()
{
double xnew[1][MAX_VARIABLE],z[1][MAX_VARIABLE];
double xold[1][MAX_VARIABLE],c[1][MAX_VARIABLE],
b[1][MAX_VARIABLE];
double lamda[MAX_VARIABLE],temp1;
double norm,sum;
double step_size=0.01,f1,f2;
double D[MAX_VARIABLE][MAX_VARIABLE],A[MAX_VARIABLE][MAX_VARIABLE];
double D_norm,d[MAX_VARIABLE],change_flag[MAX_VARIABLE];
struct
{
double proj[MAX_VARIABLE];
}s[MAX_VARIABLE];
int stage=0,l,p,n;
int success[MAX_VARIABLE],failure[MAX_VARIABLE],
change_stage_flag;
int min_flag,n_function,j,point,search_dir;

if((out=fopen("rosen.out","w"))==NULL)
{
printf("\7UNABLE TO OPEN OUTPUT FILE\n");
exit(1);
}

initiate();

min_flag=1;
if(min_flag=='2')
min_flag=0;
reenter:
n_function = 1 ;
if((n_function>no_test_function)||(n_function<=

```

```

    0))
{
if(no_test_function==0)
printf("No test Function available \n");
exit(1);
goto reenter;
}
rosenbrock=function[n_function]; /* pointer assignment */
n=dimension[n_function];
for(i=0 ; i<n ; ++i)
{
x[i][0] = 0.0 ;
}

for(i=0 ; i<n ; ++i)
for(j=0 ; j<n ; ++j)
{
if(i==j)
s[i].proj[j]=1.0;
else
s[i].proj[j]=0;
} /* selection of initial direction as principal coordinates dir */

step_size = 0.001;
stage=0;
point=0;
times_function_calculated=0;
for(j=0 ; j<n ; ++j)
xnew[0][j]=x[j][0];
do
{
++stage;
for(j=0 ; j<n ; ++j) {
t[0][j]=x[j][point]; /* Point may be 0 or anything */
}
for(i=0 ; i<n ; ++i)
lamda[i]=step_size;
search_dir=0;
for(i=0 ; i<n ; ++i)
{
change_flag[i]=0;
success[i]=0;
failure[i]=0;
d[i]=0;
}

do
{
for(j=0 ; j<n ; ++j)
{
c[0][j]=s[search_dir].proj[j];
xold[0][j]=xnew[0][j];
}
scalar_mult(1,n,c,b,lamda[search_dir]);

```

```
sum_matrix(1,n,xold,b,xnew);
f1=(*rosenbrock)(xold);
f2=(*rosenbrock)(xnew);
if(!min_flag)
{
f1=-f1;
f2=-f2;
}
if(f2>f1)          /* Failure occur */
{
++failure[search_dir];
for(j=0 ; j<n ; ++j)
xnew[0][j]=xold[0][j];
lamda[search_dir]*=-0.5;
}
else
{
++success[search_dir];
++point;
d[search_dir]+=lamda[search_dir];
for(j=0 ; j<n ; ++j)
x[j][point]=xnew[0][j];
lamda[search_dir]*=3.0;
}
++search_dir;
if(search_dir==n)
search_dir-=n;
if(point==MAX_VARIABLE-1)
point--=(MAX_VARIABLE-1);
for(i=0 ; i<n ; ++i)
if(success[i]!=0 && failure[i]!=
0)
change_flag[i]=
1;
change_stage_flag=1;
for(i=0 ; i<n ; ++i)
if(change_stage_flag)
if(change_flag[i]==
0)
/* code folded from here */
change_stage_flag=
0;
/* unfolding */
}while(change_stage_flag==0);
norm=0;
for(j=0 ; j<n ; ++j)
{
temp1=xnew[0][j]-t[0][j];
temp1*=temp1;
norm+=temp1;
}
norm=sqrt(norm);
if(norm<=TOLERANCE) goto skip;
```

```

for(l=0 ; l<n ; ++l)
{
for(j=0 ; j<n ; ++j)
A[j][l]=0;
for(i=1 ; i<n ; ++i)
{
for(p=0 ; p<n ; ++p)
c[0][p]=s[i].proj[p];
scalar_mult(1,n,c,b,d[i]);
for(j=0 ; j<n ; ++j)
A[j][l]+=b[0][j];
}
}

for(i=0 ; i<n ; ++i)
{
for(j=0 ; j<n ; ++j)
temp_vector[0][j]=0;
for(l=0 ; l<i ; ++l)
{
sum=0.0;
for(j=0 ; j<n ; ++j)
sum+=s[l].proj[j]*
    A[j][i];
for(p=0 ; p<n ; ++p)
c[0][p]=s[l].proj[p];
scalar_mult(1,n,c,b,sum);
for(j=0 ; j<n ; ++j)
temp_vector[0][j]+=
    b[0][j];
}
D_norm=0;
for(p=0 ; p<n ; ++p)
{
D[i][p]=A[p][i]-temp_vector[0][p];
D_norm+=D[i][p]*D[i][p];
}
D_norm=sqrt(D_norm);
for(p=0 ; p<n ; ++p)
s[i].proj[p]=D[i][p]/D_norm;
}
skip:
printf("");
}while(norm>=TOLERANCE);

for(i=0 ; i<n ; ++i)
temp_vector[0][i]=xnew[0][i];
}

```

Appendix B

Program Listing of the Simplified Model

```
/* smain.c */

#include <stdio.h>
#include <strings.h>
#include <string.h>
#include <math.h>
#include "rosen.h"
#include "smply.c"
#include "rosen.c"
#include "sol_input.c"
#include "seq1.c"

extern double avg_error(double) ;
char *sol_fn = "sol.out";

void main()
{
    int i, k = 0, mr_max = 1 ;
    int opt_mr, opt_eos ;
    double dif, pred_value ;
    extern int no_variables;
    int variable;
    extern char *dat;
    extern double (*rosenbrock)(), temp_vector[][MAX_VARIABLE];
    double Avg_Err, *pred_y2 = vector(6) ;
    double (*fp_prdeql)() ;
    char *opt_mr_name, *opt_eos_name ;
    char *mr_name, *mod_fn = "mod.out" ;
    char *dat_fn = "datanames" ;
    char *prsnt_mr[] = {
        "QMR_&_QMR",
        "CMR_&_QMR",
        "QMR_&_LMR",
```

B Program Listing of the Simplified Model

```
"CMR_&_LMR",  
"Wong_&_Sand" };
```

```
FILE *mod_fp ;
```

```
if ( (sol_fp = fopen(sol_fn, "at")  
    )  
    == NULL )  
{  
fprintf(stderr, "Error opening file: %s\n",  
    sol_fn);  
exit(1);  
}
```

```
if ( (mod_fp = fopen(mod_fn, "at")  
    )  
    == NULL )  
{  
fprintf(stderr, "Error opening file: %s\n",  
    mod_fn);  
exit(1);  
}
```

```
if ( (dat_fp = fopen(dat_fn, "rt")  
    )  
    == NULL )  
{  
fprintf(stderr, "fopen failure : %s\n",  
    dat_fn);  
exit(1);  
}
```

```
for (k = 1; k <= 7; k++)  
{
```

```
/*tpy_input() */  
system_input();
```

```
fprintf(sol_fp, "\n\t*** S Y S T E M ***\n");  
fprintf(sol_fp, "\n*%s\t*%s\t*%s\n",  
    fn2, fn1,  
    Pydat);  
fprintf(mod_fp, "\n%s\t%s\t%s\t", fn2,  
    fn1, Pydat);  
fprintf(sol_fp, "\nTemperature : %lf\tNo. of Data points : %d\n",  
    t, N_exp);  
fprintf(mod_fp, "%.2lf\t%d", t, N_exp);  
/*re: */
```

```
/*printf("\nEnter the no of Variables :");
scanf("%d",&no_variables);*/
no_variables = 2;

if (no_variables) {
fprintf(sol_fp, "\n\nThe no. of OPTIMIZATION parameters : %d",
    no_variables);
for (mr = 1; mr <= mr_max; mr++)
    {
mr_name = prsnt_mr[--mr];
mr++;
fprintf(sol_fp, "\n\nMixing Rule for \"a\" & \"b\" is %s",
    mr_name);
for (eos = 2; eos <= 4;
    eos++) {
Select_EoS(eos);
Calc_Constants(eos);
rosen();
Avg_Err = (((*rosenbrock)(temp_vector)));

fprintf(sol_fp,
    "\n%s---",
    dat);
fprintf(sol_fp,
    "\tkij = %.4lf\tlij = %.4lf\tAvg_Err = %.2lf",
    temp_vector[0][0],
    temp_vector[0][1],
    Avg_Err);
fprintf(mod_fp,
    "\t%.4lf\t%.4lf\t%.2lf",
    temp_vector[0][0],
    temp_vector[0][1],
    Avg_Err * 100.0);

printf("\n**AVG. ERROR : %lf\n",
    (((*rosenbrock)(temp_vector)))
    *100.0);

}
}
/* goto re;*/
}
}

/*eql(temp_vector) ;*/
}

/* smply.c */
```



```
#include "solblty.h"
#include "mix_rule.c"

/* Calculation of a,b */
void calc_const(eos, tc, pc, accentric, t, omega_a, omega_b,
a, b)
int eos;
double tc, pc, accentric, t, omega_a, omega_b, *a, *b;
{
double tr;
double alpha, beta;
tr = t / tc;
switch (eos) {
case 1:
*a = omega_a * R * R * tc * tc / pc ;
*b = omega_b * R * tc / pc ;
break;

case 2:
*a = omega_a * R * R * pow(tc, 2.0) / pc
;
beta = 0.480 + 1.574 * accentric - 0.176
*accentric * accentric ;
alpha = 1.0 + beta * (1.0 - sqrt(tr)) ;
alpha *= alpha;
*b = omega_b * R * tc / pc ;
break;

case 3:
*a = omega_a * R * R * tc * tc / pc ;
beta = 0.37464 + 1.54226 * accentric -
0.26992 * accentric * accentric ;
alpha = 1.0 + beta * (1.0 - sqrt(tr)) ;
alpha *= alpha;
*a *= alpha;
*b = omega_b * R * tc / pc ;
break;

case 4:
*a = omega_a * R * R * pow(tc, 2.5) / (pc
*sqrt(t)) ;
*b = omega_b * R * tc / pc ;
break;
default :
break;
}
}

/* Schmitt-Wenzel EOS in 'z' form */

double f(A, B, z, u, w)
```

```
int u, w;
double A, B, z;
{
return z * z * z + (u * B - B - 1.0) * z * z +
    (w * B * B - u * B * B - u * B + A) * z - (A *
    B + w * B * B + w * B * B * B);
}

/* Differential of S-W-Eos */

double df(A, B, z, u, w)
int u, w;
double A, B, z;
{
return 3.0 * z * z + 2.0 * (u * B - B - 1.0) *
    z + (w * B * B - u * B * B - u * B + A);
}

/* Calculation "vap-z" & vap-vol. */

double calc_z(u, w, p, t, a, b, guess)
int u, w;
double p, t, a, b, guess;
{
double z, diff, dfz, fz, v;
double A, B;
int i = 0;

B = b * p / (R * t);
A = a * p / (R * R * t * t);
z = guess;
do
{
dfz = df(A, B, z, u, w);
/*if (dfz<TOL)
{
    fprintf(stderr,"Denom < TOL in newton raphson for Z");
    exit(3);
}*/
fz = f(A, B, z, u, w);
guess = z - fz / dfz;
diff = fabs(z - guess);
z = guess;
} while (i++ < ITER && diff > TOL);
if (diff > TOL) /*not converged*/
return - 100;
/*printf("\nz=%lf",z);*/
return z;
}
```

```
double fug_PR_infnty(double z, double al, double bt, double
A,
double B)
```

```
{

double t1, t2, t3, t4, D, r1, r2 ;

D = u * u - 4 * w ;
r1 = (-u + sqrt(D)) / 2 ;
r2 = (-u - sqrt(D)) / 2 ;
t1 = bt * (z - 1.0) ;
t2 = log(z - B) ;
t3 = A * (2 * al - bt) / (sqrt(D) * B) ;
t4 = log( (z - r2 * B) / (z - r1 * B) ) ;

return exp( t1 - t2 - t3 * t4 ) ;
}
```

```
double solblty(t, p, fug2)
```

```
double t, p, fug2 ;
```

```
{
double pf ;

pf = exp(Vs * (p - p_vap) / (R * t));

return p_vap * pf / (p * fug2);
}
```

```
double func(x)
```

```
double x[][MAX_VARIABLE];
```

```
{

int i, j, stat ;
double guess = 1.0, y2_cal[N], diff[N], sum = 0.0
;
double z_PR, fug2_PR_inf, OF, al, bt , A , B ;

al = x[0][0] ;

bt = x[0][1] ;

for (i = 0; i < N_exp; i++) {

A = al * P[i] / sq(R * t) ;
```

```
B = b1 * P[i] / (R * t) ;

z_PR = calc_z(u, w, P[i], t, a1, b1, guess);

fug2_PR_inf = fug_PR_infnty(z_PR, a1, bt,
    A, B) ;

y2_cal[i] = solblty(t, P[i], fug2_PR_inf);
}

for (i = 0; i < N_exp; i++) {
diff[i] = sq( (y2_cal[i] - y2_exp[i]) /
    y2_exp[i] );
sum += diff[i] ;
}

OF = sqrt(sum / N_exp) ;

return OF ;
}
```

```
double avg_error(OF)
double OF;
{
int i;
double sum = 0.0;
for (i = 0; i < N_exp; i++) {
sum += y2_exp[i];
}

return OF * N_exp * 100.0 / sum ;
}
```

```
void Select_EoS(eos)
{
switch (eos) {
case 1:
dat = fn[0];
break;
case 2:
dat = fn[1];
break;
case 3:
dat = fn[2];
break;
case 4:
dat = fn[3];
break;
default:
```

```
break;
}

if ( (fp = fopen(dat, "rt")) == NULL ) {
fprintf(stderr, "fopen failure : %s\n",
    dat);
exit(1);
}
fscanf(fp, "%d%d%lf%lf", &u, &w, &omega_a, &omega_b);
fclose(fp);
}

void Calc_Constants(eos)
{
calc_const(eos, tc1, pc1, accentric1, t, omega_a,
    omega_b, &a1, &b1);
calc_const(eos, tc2, pc2, accentric2, t, omega_a,
    omega_b, &a2, &b2);
}

/* prdeql.c */

/*.. Predict Equilibria ..*/
extern double avgerror(double) ;

void predict_eq1(int opt_mr, int opt_eos, double x[][MAX_VARIABLE])
{
int i, j, stat ;
double guess = 1.0, y2_cal[N], diff[N], sum = 0.0
    ;
double z_PR, fug2_PR_inf, OF, al, bt , A , B ;
double T_pred, P_exp[N] ;
char Pred_Pydat[20] , *eq1 = "eq1.out";
FILE * pred_fp, *eq1_fp ;

/*retry :fprintf(stderr, "\nEnter *T-P-y* Filename : ");
    scanf("%s", Pred_Pydat);*/
if ( (pred_fp = fopen(Pydat, "rt")) == NULL ) {
fprintf(stderr, "fopen failure : %s\n",
    Pydat);
}
fscanf(pred_fp, "%lf%d", &T_pred, &N_exp);
for (i = 0; i < N_exp; i++) {
fscanf(pred_fp, "%lf%lf", &P_exp[i], &y2_exp[i]);
```

```

}
fclose(pred_fp);
fprintf(sol_fp, "\n\n** Prediction at %.2lfK **",
        T_pred);
p_vap = vap_pr(T_pred) ;
mr = opt_mr ;
eos = opt_eos ;
Select_EoS(eos) ;
Calc_Constants(eos) ;

a1 = x[0][0] ;

bt = x[0][1] ;

for (i = 0; i < N_exp; i++) {

A = a1 * P[i] / sq(R * t) ;

B = b1 * P[i] / (R * t) ;

z_PR = calc_z(u, w, P[i], t, a1, b1, guess);

fug2_PR_inf = fug_PR_infnty(z_PR, a1, bt,
        A, B) ;

y2_cal[i] = solblty(t, P[i], fug2_PR_inf);
}

for (i = 0; i < N_exp; i++) {
diff[i] = sq( (y2_cal[i] - y2_exp[i]) /
        y2_exp[i] );
sum += diff[i] ;
}

OF = sqrt(sum / N_exp) ;

printf("\n\nAvg. Error of Prediction : %lf", OF) ;
printf("\n\nCalculated \t Experimental") ;
for (i = 0; i < N_exp; i++) {
printf("\n%e \t %e", y2_cal[i], y2_exp[i]) ;
}
fprintf(sol_fp, "\n\nCalculated \t Experimental") ;
fprintf(sol_fp, "\n\n-----");
for (i = 0; i < N_exp; i++) {
fprintf(sol_fp, "\n%e \t %e", y2_cal[i],
        y2_exp[i]) ;
}
fprintf(sol_fp, "\n\nAvg. Error of Prediction : %lf",
        OF) ;
fprintf(sol_fp, "\n\n*****\n") ;
}

```

```
/* Sol_input.c */

double vap_pr(t)
double t;
{

if (!strcmp(fn2, "benzoic"))
return exp(pA2 - pB2 / (t + pC2)) ;

else if (!strcmp(fn2, "anthracn"))
return (0.0013332) * exp( (pA2 - pB2
/
(t + pC2) ) );

else if ((!strcmp(fn2, "acridn")) || (!strcmp(fn2,
"naptqn")))
return 1.0e-5 * pow(10.0, (pA2 - pB2 /
(t + pC2)));

else
return pow(10.0, (pA2 - pB2 / (t + pC2)));

}

/*
void tpy_input()
{
fprintf(stderr, "\nEnter *T-P-y* Filename : ");
scanf("%s", Pydat);
if ( (fd=fopen(Pydat, "rt")) == NULL )
{
fprintf(stderr, "fopen failure : %s\n", Pydat);
exit(1);
}
fscanf(fd, "%lf %d %p_vap", &t, &N_exp, &p_vap);
for(i=0; i< N_exp; i++)
{
fscanf(fd, "%lf%lf", &P[i], &y2_exp[i]);
}
fclose(fd);
}
*/

void system_input()
{
int opt ;

/*fprintf(stderr, "\nEnter *SOLVENT* Filename: ");
scanf("%s", fn1);*/
```

```
fscanf(dat_fp, "%s %s %s\n", fn2, fn1, Pydat);

if ( (fq = fopen(fn1, "rt")) == NULL ) {
fprintf(stderr, "fopen failure : %s\n",
    fn1);
exit(1);
}
fscanf(fq, "%lf%lf%lf%lf", &mi, &tc1, &pc1, &accentric1);
fclose(fq);

/*fprintf(stderr, "\nEnter *T-P-y* Filename : ");
scanf("%s", Pydat);*/
if ( (fd = fopen(Pydat, "rt")) == NULL ) {
fprintf(stderr, "fopen failure : %s\n",
    Pydat);
exit(1);
}
fscanf(fd, "%lf %d %lf", &t, &N_exp, &p_vap);
for (i = 0; i < N_exp; i++) {
fscanf(fd, "%lf%lf", &P[i], &y2_exp[i]);
}
fclose(fd);

/*fprintf(stderr, "\nEnter *SOLUTE* Filename : ");
scanf("%s", fn2);*/
if ( (fr = fopen(fn2, "rt")) == NULL ) {
fprintf(stderr, "fopen failure : %s\n",
    fn2);
exit(1);
}
/*printf("\nDo you have p_sat (1/0):");
scanf("%d", &opt);*/
opt = 1 ;
if (opt) {
/*printf("\nEnter p-sat in bars :");
scanf("%lf", &p_vap);*/
fscanf(fr, "%lf%lf%lf%lf%lf", &mj, &tc2,
    &pc2, &accentric2, &Vs);
} else {
fscanf(fr, "%lf%lf%lf%lf%lf%lf%lf%lf",
    &mj, &tc2, &pc2, &accentric2, &Vs, &pA2,
    &pB2, &pC2);
p_vap = vap_pr(t);
}
fclose(fr);
}
```